Unclassified

FG

Project Report

PPP-74
(PRESS)

T. H. Einstein

Introduction to the PRESS Real-Time Program

5 October 1967
Reissued
7 March 1969

Prepared for the Advanced Research Projects Agency
under Electronic Systems Division Contract AF 19(628)-5167 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts

Unclassified

BEST
AVAILABLE COPY

Unclassified

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This research is a part of Project DEFENDER, which is sponsored by the U.S. Advanced Research Projects Agency of the Department of Defense; it is supported by ARPA under Air Force Contract AF 19(628)-5167 (ARPA Order 600).

Unclassified

0

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

6

INTRODUCTION TO THE PRESS REAL-TIME PROGRAM.

10

T. H. EINSTEIN

Division 3

9 14

PROJECT REP[redacted]RT, PPP-74 ~~(PRESS)~~

11 5 OCT[redacted]67

21 REISSUED 7 MAR[redacted]69.

12 65p.

15 AF19(628)-5167, ARPA Order-60

LEXINGTON MASSACHUSETTS

MISSING PAGE
NUMBERS ARE BLANK
AND WERE NOT
FILMED

ABSTRACT

This document gives an introductory description of the function and operation of the PRESS Real-Time Program (RTP). The RTP operates in a 7094 computer and is used to record data and to control the TRADEX radar and PRESS optical sensors at the PRESS Field Station at Roi-Namur Island, Kwajalein Atoll, Marshall Islands. The RTP is subdivided into 29 semi-autonomous subprograms, and the function and operation of each of these subprograms is briefly described. The emphasis in these descriptions is on the operation of the RTP viewed as a system of component subprograms, and on the data communication between the RTP and the external I/O devices connected to the computer.

CONTENTS

FOREWORD

This document gives an introductory description of the PRESS Real-Time Program (RTP). The intent of the description given herein is to acquaint the reader with the functions performed by the RTP and to present the fundamentals of the program's operation. The RTP itself is subdivided into 29 principal subprograms, and a very brief description of the function and operation of each of these subprograms is included. However, a detailed description of the actual coding of the component subprograms has been purposely omitted since the author's intent is to give the reader a view of the over-all operation of the RTP, rather than detailed operational descriptions of the individual subprograms.

Attempting to document the PRESS RTP has proved to be a somewhat frustrating task because the RTP, like any other PRESS subsystem, is in a continuing state of development. Consequently, this document suffers from the shortcoming that it is, to a certain extent, already obsolete. Nevertheless, despite modifications that have been and are being continually made to upgrade the program, the basic functions and structure of the program have remained essentially unchanged over the past four years. The description presented here is that of the state of the program as it existed approximately in the Fall of 1966, shortly after the author's departure from the PRESS Field Station (PFS) following a 32-month tour of duty. Despite the considerable number of modifications made to the program since that time, which are not reflected in the present description, it is hoped that this document will nevertheless be useful in acquainting the reader with the general purpose and functions of the program, and will serve as an introductory guide to its principles of operation.

This Foreword would not be complete without a brief description of the history of the RTP. The basic design of the present form of the program was originally developed in 1963 by K. E. Ralston and J. E. Morriello. Their basic design provided a modular program structure that greatly facilitated the implementation of future additions and modifications to the program. The fact that the original program structure has indeed remained essentially unchanged during the past four years, and has been able to accommodate the many major additions and modifications made during that period, stands as a tribute to the ingenuity and foresight of the original designers. During the period 1964 – 1967, the following persons contributed to the further development of the program: Dr. H. E. Frachtman, G. I. Tabasky, and the author. As of this writing (August 1967), the program is still being upgraded and its development is being continued under the guidance of R. Teoste, the author's successor at PFS.

INTRODUCTION TO THE PRESS REAL-TIME PROGRAM

1.0 INTRODUCTION

The PRESS Real-Time Program (RTP) is a computer program for coordinating and controlling the operation of various PRESS sensor subsystems, and for recording the data collected by these sensors in real-time. The data flow between the various sensors and the PRESS RTP resident in the 7094-II computer is shown in Fig. 1.

1.1 FUNCTIONS

The principal functions performed by the PRESS RTP are:

(1) Support and control of TRADEX radar, including

(a) Initial target acquisition and target reacquisition,

(b) Generation of feed-forward rate commands to reduce servo lag errors while tracking,

(c) Control of TRADEX prf to keep the target return out of the ambiguous range and doppler clutter (2-D prf generation).

(2) Pointing ground and airborne optics sensors at the target(s) being tracked by TRADEX.

(3) Real-time recording of data from all PRESS sensors that are tied to the computer.

(4) Providing real-time display of aircraft and target positions on a geographic overlay plot on Milgo plotting boards.

The primary purpose of the RTP is to coordinate and control the operation of the PRESS sensors, and to record data during ballistic vehicle flight test missions. These test missions are called "PRESS Tests." During a PRESS Test, exoatmospheric and re-entry physics data are collected by the sensors on the ballistic vehicle complex that is being flown for the test. However, the RTP is also used to support the following auxiliary operations, which are performed primarily for over-all system check-out and diagnosis:

(1) TRADEX IF Tape Playbacks,

(2) TRADEX Satellite Tracks,

(3) PRESS Aircraft Tracking and AOCS Tests,

(4) Balloon Tracks and Radar Calibration,

(5) Generation of Celestial Tracking Data to point optics instruments at stars,

(6) Diagnostic check-out of data links between the computer and the PRESS sensors,

(7) Computer Tape Playbacks to simulate live data from sensors for evaluation and check-out of modifications to the RTP.

1.2 PROGRAM STRUCTURE

The RTP actually consists of 29 separate, principal subprograms – each subprogram performing a unique function within the system. In addition to these 29 principal subprograms, the RTP also contains about a dozen minor subprograms that perform miscellaneous support functions,

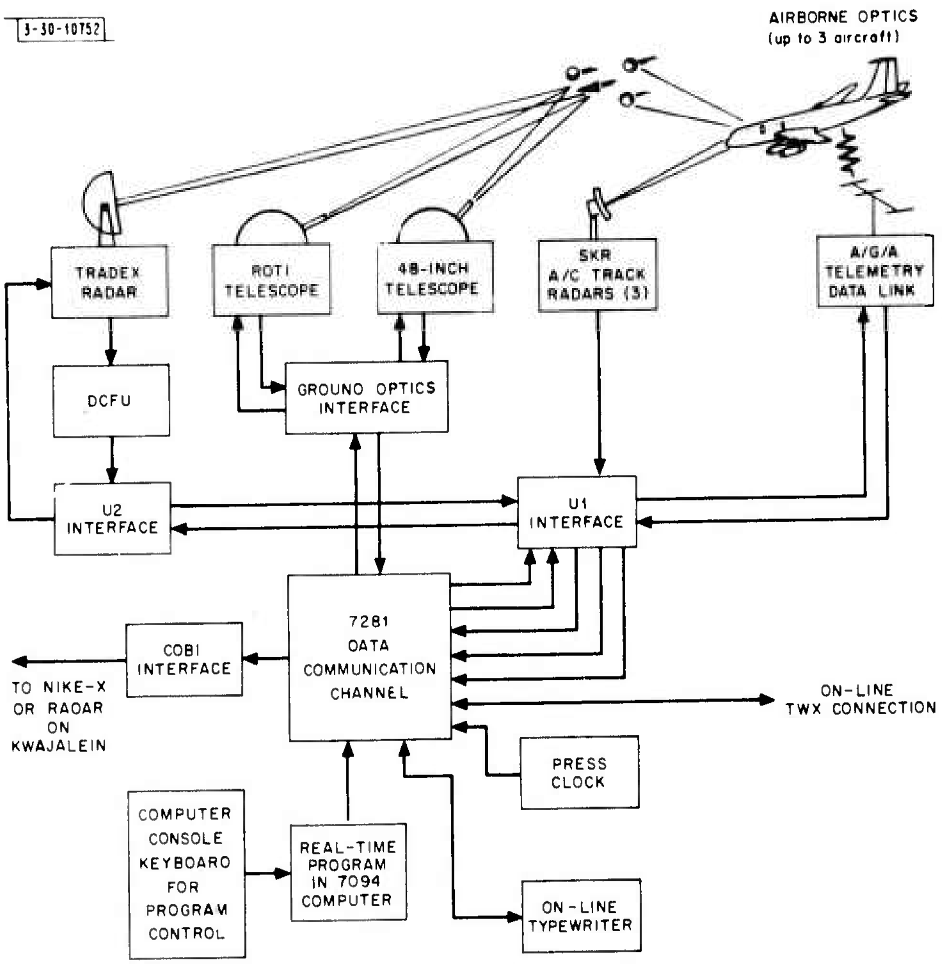

Fig. 1. PRESS sensor control system.

such as computation of trigonometric functions, refraction correction, table look-up, and printer line image conversions.

The basic structure of the program is centered around the computation of pointing data for the PRESS sensors, these pointing data being computed from target trajectories which are derived from TRADEX radar tracking data. For example, during the course of a test mission, as TRADEX tracks various targets in the target complex (only a single target being tracked at any given time), trajectory estimates for each target tracked are derived from the TRADEX metric tracking measurements. These target trajectory estimates, called "track files," become the basis for the computation of pointing data for the other PRESS sensors, thus enabling these sensors to be pointed at any target being tracked, or having been previously tracked, by TRADEX. The specific content of and role played by these track files in the RTP is described in Sec. 2.0. The performance of the various program functions associated with the generation of these track files, and the subsequent computation by sensor pointing data, are divided among several of the individual principal subprograms. For example, one of the principal subprograms decodes and corrects the TRADEX track data used to generate the track files. The actual smoothing of these data into the track files is performed by another subprogram. Still another subprogram performs the extrapolation or trajectory integration of the track files. Separate subprograms then compute the pointing commands for each of the individual PRESS sensors. A complete description of the operation and function of each of the 29 principal subprograms is given in Sec. 7.0.

The principal subprograms exchange data with one another through the COMMON data storage area of the program. The COMMON data storage area is a reserved block of about 2000 storage locations in the upper memory of the computer. This block of 2000 storage locations is accessible by every subprogram through a COMMON storage address list which is assembled with each subprogram. With a few exceptions, the subprograms communicate with each other exclusively through the COMMON area, but are otherwise independent. The principal exceptions are the various special control programs associated with the on-line teletype, the typewriter, the on-line printer, and the Milgo plot boards. The primary means of data communication with these subprograms is through the subprogram calling sequences, rather than through COMMON. The organization of the RTP into subprograms, which are largely independent of one another as described above, results in a modular and highly flexible program structure. For example, because of the above structure, new subprograms are easily added to the RTP, and existing subprograms may be individually modified without affecting the operation of the remainder of the system. The modularity of the system also greatly simplifies the process of debugging modifications and additions to the RTP.

Since the primary data communication rate with the external sensors is ten messages per second, the RTP also operates in a 100-msec cycle. That is, the execution of the RTP is repeated ten times each second. Each execution of the RTP every tenth second is defined as a program cycle. The operation of the RTP during each of the program cycles is identical, except that those operations which need be executed less frequently than 10/sec are only performed during particular program cycles each second. For example, certain operations associated with writing the computer output tape are performed only once a second during the 0.0- and 0.2-sec program cycles. Also, the TRADEX metric data are sampled for inclusion in the smoothed target trajectories only twice a second on the 0.1- and 0.6-sec program cycles. Other special

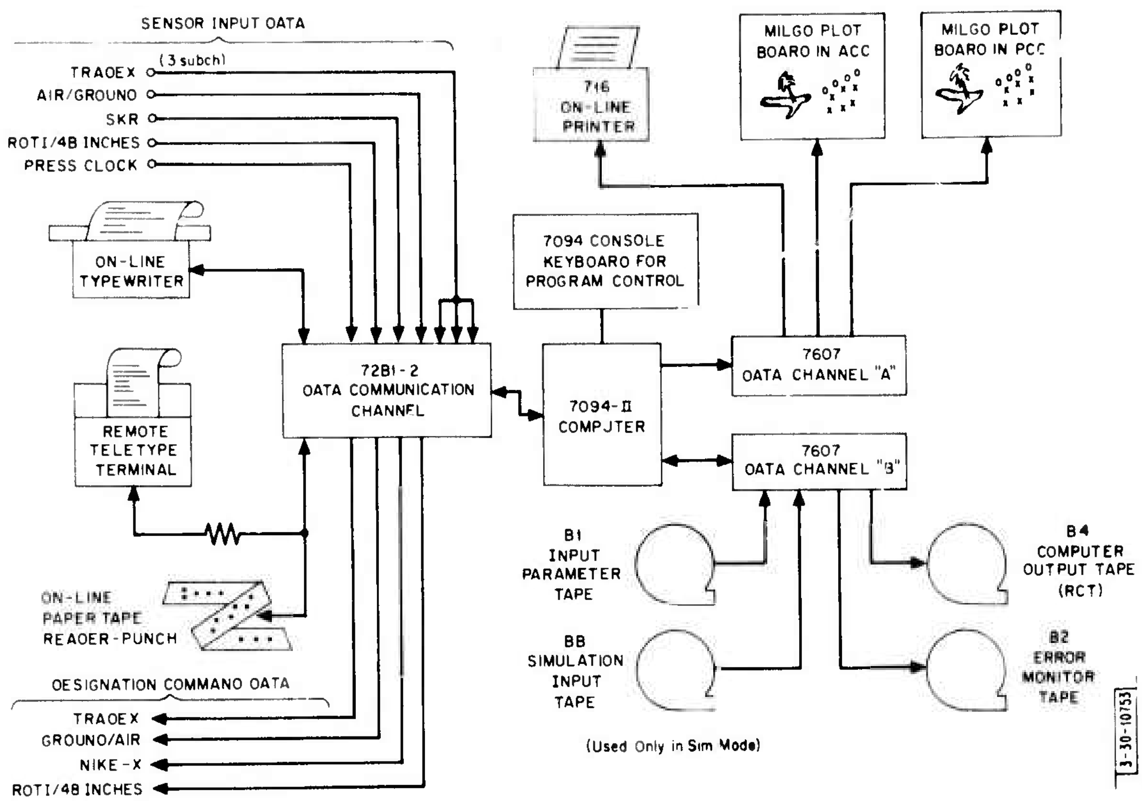

Fig. 2. PRESS RTP external I/O connection diagram.

operations that are performed less frequently than once per program cycle, such as the re-entry prediction computations and plotting, are executed on a time-share basis over a period of several program cycles.

1.3 EXTERNAL I/O

An interconnection diagram showing the various external data devices connected to the RTP in the 7094-II is shown in Fig. 2. A description of these devices and their functions is given in the following paragraphs.

1.3.1 7281 Data Communication Channel

The 7281 Data Communication Channel is used to exchange data between the program and the external digital data links to the PRESS sensors. The 7281 itself presently consists of 14 independent subchannels and a multiplexor. The 7281 has a maximum capacity of 32 subchannels and additional subchannels may be added, as required, up to that limit. The 7281 multiplexor time-shares the communication of data between the various subchannels and the computer memory. These subchannels are designated as input, output, or duplex (input + output), and each services a separate data link. For example, for communication with the optics sensors in the aircraft, the output commands from the computer to the aircraft are transmitted through the ground-to-air (G/A) output subchannel, and the sensor position data in the aircraft are sent back to the computer through the A/G input subchannel. Only the on-line typewriter and teletype connection are serviced through duplex subchannels. The subchannels are connected to the external data links through data interface equipment. The interface equipment converts the logic voltage levels of the data links to those required by the subchannels and generates the timing signals for transferring data through the subchannels. The subchannels presently installed in the 7281 are listed in Table I.

1.3.2 Magnetic Tapes

Data are recorded by the program on two magnetic tape units that are connected to the computer through 7607 data Channel "B." The first of these is the Real-Time Computer Tape (RCT), on which are recorded (1) all the data entering the computer from the PRESS sensors through the 7281 input subchannels, (2) the Target Track Files and sensor directing data computed by the program, and (3) program status information. The program moves the continuously incoming data from the 7281 input subchannels, and the program computed data to be recorded, to an output tape buffer from which all the data are written out onto the RCT once every second. The 7281 input data are recorded exactly as received; no processing is performed by the program on the 7281 data recorded on the RCT.

The other tape is the ERROR Monitor Output Tape (EMT), on which are recorded any error conditions or other anomalies sensed by the program. The error conditions recorded are those due to malfunctions in the data links and other I/O equipment attached to the computer, and anomalies in the operation of the program itself. The contents of this tape are used as a diagnostic tool in debugging system malfunctions. Data describing the nature of the error are recorded on this tape whenever an error is sensed by the program.

The program also initially reads an input parameter tape. This tape is only read once at the time that the program is started. The input parameter tape contains operating parameters,

TABLE 1 INSTALLED 7281 SUBCHANNELS		
Subchannel No.	Subchannel Name	Word Length and Type
(1)	TRADEX Directing	16-bit serial output
(2)	G/A Directing	16-bit serial output
(3)	A/G Input	32-bit serial output
(4)	NIKE-X Directing	16-bit serial output
(5)	Ground Optics Directing	16-bit serial output
(6)	Ground Optics Read-back	32-bit serial input
(7)	TRADEX Input No. 1	36-bit parallel input
(8)	TRADEX Input No. 2	36-bit parallel input
(9)	SKR Input	16-bit parallel input
(10)	TRADEX Gain Input	32-bit serial input
(11)	Open	
(12)	Open	
(13)	Interval Timer	16-bit counter set by program that is decremented by external demand pulses
(14)	PRESS Clock	36-bit parallel input
(15)	Typewriter	7-bit serial half-duplex
(16)	Teletype	5-bit serial half-duplex

such as target drag characteristics, atmospheric property tables, and data conversion constants that are required by the RTP. This input parameter tape is prepared by a separate input program from data on punched cards.

1.3.3 On-Line Printer and Plot Boards

The on-line printer attached to the computer is used by the program to print messages to the computer operator and RTP operations director. These messages include those operational error alarms and program status information which require the immediate attention of the RTP operations director. Examples of messages displayed on the on-line printer are alarms indicating invalid manual control operations and an indication of the arrival in the computer of data messages from the on-line teletype link.

The program also drives two Milgo plot boards that are used to display radar target and aircraft positions on a 300- × 300-mile geographic overlay of the Kwajalein area. Points are plotted on these boards at the rate of about three points every ten seconds. Both the on-line printer and the Milgo plot boards are attached to the computer through the 7607 Data Channel "A."

1.3.4 On-Line Typewriter

An on-line typewriter is attached to the computer through the typewriter subchannel of the 7281. The typewriter provides a means of manual communication with the program and may be used to query the contents of any storage location in the computer or to enter data into the computer while the program is operating. At present, its principal use during a PRESS mission is

to enter the time of target launch into the program upon receipt of that time from the launch agency. The capabilities of the typewriter also make it an extremely useful diagnostic tool for checking out program modifications.

1.3.5 On-Line Teletype Connection

An on-line teletype connection with the computer is provided through the teletype subchannel of the 7281. The teletype subchannel is connected to either a paper tape reader-punch unit or to the external radio-teletype lines by means of an external manual switch. The subchannel provides either input or output communication, under program control, with the external devices. Presently, this facility is used to enter the launch agency target acquisition message (offset message) into the computer, directly as it is received via the radio-teletype link, and to generate an acquisition message for the ARIS tracking ships operating in the local area.

1.4 PROGRAM OPERATING MODES

The RTP may be made to operate in any one of the following three modes: Live, Simulation, or IF Tape Playback (IFPB). As will be seen, the differences in these various modes of operation are in the sources of input data and timing used by the program. The mode of operation desired is manually selected by the computer operator through the computer console keys. A description of each mode follows.

1.4.1 Live Mode

Live Mode is the normal mode of operation for the RTP. In Live Mode Operation, sensor input data enter the computer through the 7281 input subchannels as described earlier, and the operation of the program (the execution of the 100-msec program cycles) is synchronized to the time from the external PRESS Master Clock. The PRESS Clock time enters the computer through the PRESS Clock subchannel of the 7281.

1.4.2 Simulation Mode

When the program operates in the Simulation Mode, all 7281 input subchannels, with the exception of the PRESS Clock subchannel, are turned off. The sensor input data to the program, in lieu of arriving through the 7281 as in Live Mode, are read from a previously recorded computer output tape (RCT). As mentioned earlier, this tape contains all the 7281 input data recorded by the program during a previous operation. The computer output tape being read during a Simulation Operation is referred to as the Simulation Input Tape to distinguish it from the computer output tape recorded during the Simulation run. In Simulation Mode, the time from the PRESS Clock is again used, but only to synchronize the operation of the program. In Simulation Mode, the actual time-of-day used by the program is taken from the data read from the Simulation Input Tape, rather than from the PRESS Clock.

The use of the designation Simulation Mode for this type of program operation should now be obvious. Since the input data and times used by the program when in Simulation Mode are those recorded during a previous operation – usually a Live Mode Operation – the program effectively replays or simulates the earlier operation. This mode of operation is used primarily as a diagnostic tool to check out additions or modifications to the program. In this way, the effect of program modifications on program performance with real data can be evaluated and compared against the results previously obtained on the same data with the unmodified program.

1.4.3 IFPB Mode

The operation of the program in IFPB Mode is similar to that in Live Mode, with the exception that program operation is synchronized to the timing signals recorded on the IF Tape, rather than to the PRESS Clock. As in Live Mode, all 7281 data input subchannels are turned on; however, in IFPB Mode, the PRESS Clock subchannel is turned off. Although all input subchannels are operative, only the data entering from the TRADEX input subchannels are actively used by the program when in IFPB Mode. The 100-pps timing pulses recorded on the IF Tape effectively cause the TRADEX Gain 7281 input subchannel to interrupt the computer every 10-msec in synchronism with these timing pulses. The resultant interrupts, occurring at a rate of 100/sec, are used to step a program generated "pseudo-clock" in synchronism with the time-of-day data recorded on the IF Tape. This program generated pseudo-clock is then used in lieu of the PRESS Clock to synchronize the operation of the program in the IFPB Mode.

1.5 PROGRAM CONTROL

Certain external manual control functions exist for governing the operation of the program. These control functions include: program mode selection (Live, Simulation, or IFPB), start or stop computer output tape recording, activate remote control panel, start or stop plotting on Milgo plotters, initiate transmission of ARIS ship acquisition message via teletype data link, inhibit use of TRADEX data by program, track file selection for computation of directing data, end program operation, and several others. All these control functions may be exercised from the manual control keyboards on the computer console. These keyboards contain a total of 42 control switches. The controls pertaining to the acceptance of TRADEX data and operation of the track files may also be exercised remotely from the remote computer control console in the TRADEX Console Room. This remote control console at TRADEX is enabled by the remote control enable key on the computer console keyboard. When remote control is enabled, only those control functions pertaining to TRADEX data and the track files are transferred from the computer console to TRADEX. All other control functions pertaining to general program operation, such as program mode selection, computer output tape recording, remote control enable, and end program operation, remain at the computer console even when TRADEX remote control is enabled.

2.0 PROGRAM GENERATED TRACK FILES

Central to an understanding of the operation of the RTP is the concept of a track file. A track file consists basically of the position and velocity components of an object in TRADEX Rectangular Coordinates at some instant of time. These position and velocity components define the trajectory or flight path of that object.

The TRADEX Rectangular Coordinate System is earth-fixed (non-inertial) with its origin at the focus of the TRADEX antenna; its z-axis is the local geodetic vertical; its x- and y-axes lie in the local tangent plane with y pointing north and x east – parallel to the earth's equatorial plane – completing the right-handed system. The position coordinates of the track files in this system are in units of ft and velocity is in units of ft/sec. There are three basic types of track files generated by the RTP: TRADEX Target Track Files, Acquisition Track Files, and Aircraft Track Files.

A track file actually consists of a block of 11 words containing the following data: time or number of track data points, target position, velocity and acceleration components, and target altitude. However, as described above, only seven of these 11 quantities, namely, time, and the target position and velocity components, are required to define a trajectory. The format for each of the three types of track files mentioned above is essentially the same and is given below.

Word No.	Data	Units	
(1)	Number of points in TDX Target TF Time of day in other TF	sec	
(2)	x-position coord of target	ft	
(3)	y-position coord of target	ft	
(4)	z-position coord of target	ft	
(5)	$\dot{x}$-velocity coord of target	ft/sec	
(6)	$\dot{y}$-velocity coord of target	ft/sec	
(7)	$\dot{z}$-velocity coord of target	ft/sec	
(8)	$\ddot{x}$-acceleration coord of target	ft/sec^2	omitted in aircraft track files
(9)	$\ddot{y}$-acceleration coord of target	ft/sec^2	
(10)	$\ddot{z}$-acceleration coord of target	ft/sec^2	
(11)	H, target altitude	ft	

Once a track file has been established, it is integrated or extrapolated in real-time by the RTP. A description of each of the three types of track files follows.

2.1 TRADEX TARGET TRACK FILES

The TRADEX Target Track Files are formed by smoothing the range, azimuth and elevation metric measurements of target position made by the TRADEX radar while tracking. Generally, the smoothing is performed by fitting a ballistic trajectory to the measurements in a least-squares sense; however, a special mode of operation exists, called "Balloon Track Mode," in which the radar measurements are least-squares fitted by a straight line instead of by a ballistic trajectory.

As the name implies, this mode of operation is used when TRADEX is tracking a nonballistic object, such as a balloon or an aircraft. The results of the trajectory fitting or smoothing process are estimates of the target's current position and velocity, and these quantities are placed into the track file. The smoothing process is performed recursively by the program so that when a new data point (measurement) is obtained, the new estimates of position and velocity are computed by using only the values of the previous estimate and the new measurement. The use of recursive trajectory fitting eliminates the need for saving previous data points for computing the least-squares fit trajectory and is computationally both simple and fast.

In addition to position and velocity, the following quantities are also placed into the TRADEX Target Track File: number of data points which have been included in that file, components of target acceleration, and altitude of the target. The acceleration components during the exoatmospheric portion of a trajectory are computed from the equations of motion for a ballistic trajectory as a function of the position and velocity components. During the atmospheric portion of the trajectory (re-entry), a drag component of acceleration is added to the values computed from the ballistic equations of motion.

The drag deceleration for the track file of the target being tracked through re-entry by the radar is computed from the radar doppler measurements. For the other track files, which are merely being extrapolated through re-entry, the drag deceleration is computed by using values of target drag coefficient prestored in a table as a function of Mach number or altitude. The altitude of a target is its local height above the reference earth ellipsoid and is computed directly from the position coordinates in the track file.

The equations for the integration of the TRADEX Track Files along a ballistic trajectory, for drag deceleration computation, target altitude computation, and recursive least-squares smoothing of TRADEX tracking measurements are given in the Appendix.

The above description has illustrated the method of forming a given TRADEX Target Track file. Presently, up to eight separate TRADEX Target Track Files may be formed by the program. Of course, TRADEX tracking measurements are filled or smoothed into only one track file at a time. The radar tracking measurements are sampled every half second for inclusion in the track file being filled. However, all established track files are extrapolated or integrated ahead in real time ten times a second, or once each program cycle. Thus, each track file is, in theory, uniquely associated with a given target that has been tracked by TRADEX at some time in the past, and at any given time the contents of each track file are the predicted positions and velocities of the corresponding target at that time. The program includes an automatic target discrimination feature which causes a new track file to be started automatically when TRADEX switches to another target. A simplified description of this feature follows. The program compares each new TRADEX data sample point with the predicted position in the track file of the target most recently tracked by the radar. If the coordinate differences between the new data point and the predicted position of that track file are less than 500 ft in range and 0.3° in angle, the program assumes that TRADEX is still tracking the same target and the new data point is smoothed into that same track file. If the position differences exceed the above criteria, the program assumes that TRADEX has switched track to another target, and the data point is used to start a new track file. However, a new track file is not considered firm until at least 20 consistent new track data points, which do not fit into the previous file, have been placed into it (10 sec of tracking data). This feature also provides for automatic rejection of excessively noisy data points from inclusion in the track files.

2.2 ACQUISITION TRACK FILES

The Acquisition Track Files contain ballistic trajectories which are integrated in real-time from a given set of initial conditions that are entered into the program as input data. At present, there are two Acquisition Track Files – Nominal and Offset. The initial conditions for the Nominal Track File consist of position and velocity components of some point on the nominal trajectory, and the time after lift-off for that trajectory point. These initial conditions are read into the program from the input parameter tape when the program is started. However, the integration of the Nominal Track File is not started until after the actual target lift-off time has been entered into the program through the on-line typewriter. The Offset Track File is similar to the Nominal, except that its initial conditions are entered through the on-line teletype link and include the actual time of the data point, rather than a time increment referenced to time of lift-off.

Logic is provided in the program to synchronize the integration of these track files with real-time. Thus, if the time-of-day appearing in the initial conditions is greater than real-time, integration of that track file is delayed until the actual time-of-day given in the initial conditions is reached. Conversely, if the time-of-day in the initial conditions is less than the real-time, that track file is integrated ahead at a rate ten times real-time to "catch-up" with the real-time.

When the altitude of an Acquisition Track File decreases below 400 kft, the aerodynamic drag on the "target" in that track file is computed by using a target drag coefficient obtained from a prestored drag table in which this drag coefficient is given as a function of target altitude or Mach number. The equations for the ballistic integration of the Acquisition Track Files are the same as those for the ballistic integration of the TRADEX Target Track Files, and are given in the Appendix.

2.3 AIRCRAFT TRACK FILES

The Aircraft Track Files are formed by smoothing the metric measurements of aircraft position made by the Station-Keeping Radar (SKR) aircraft tracking radars. In a manner similar to the ballistic trajectory fit of TRADEX Target Track Files, the Aircraft Track Files are formed by recursively fitting a straight line in a least-squares sense to the aircraft position measurements made by the SKRs. There is a total of three Aircraft Track Files – each being associated with one of the three SKR aircraft tracking radars. These track files have essentially the same format as the TRADEX Target Track Files – consisting of estimates of target position, velocity, and altitude – except that target acceleration and number of data points are omitted. The straight-line fit smoothing technique is based on the assumption that the aircraft flies a steady course and, therefore, the track file acceleration components are implicitly zero. The word in the Aircraft Track File corresponding to the one in the TRADEX Target Track File containing the number of points, contains instead time-of-day.

The SKR data are sampled for smoothing into the Aircraft Track File ten times a second, or once every program cycle, and the track files are extrapolated at the same rate in real-time. Unlike the TRADEX Target Track Files, no target discrimination logic is provided for the Aircraft Track Files because it is assumed that each SKR will always track the same aircraft. However, logic is provided to reject excessively noisy data points.

The quality of the Aircraft Track Files is enhanced through the use of aircraft altitude measurements made by radar altimeters in the aircraft. These data are transmitted to the SKR data link through the air-to-ground telemetry data link. These altimeter measurements of aircraft altitude are highly accurate, and therefore, when available, they are used in lieu of the less accurate SKR elevation measurements to compute the aircraft position and velocity estimates in the track file.

3.0 GENERATION OF SENSOR DIRECTING DATA

Pointing or directing data for the PRESS sensors are computed from the target position and velocity estimates in the track files. These pointing data may be computed from any of the eight TRADEX Target Track Files, either of the Acquisition Track Files, or from Aircraft Track File No. 3. At present, a single track file from the above set is selected for computation of directing data, and the pointing data for all the sensors are computed from that track file. This selection may be made from the computer or remote control consoles. Consequently, every sensor that slaves to computer-generated pointing data is pointed at the same object. The track file selected for computation of directing data is referred to as the "Directing Track File." However, extensive program and external control modifications are currently under way to enable the pointing data for each sensor to be computed from a separate track file if desired. When this modification is completed, any of the existing track files may be separately selected for each sensor so that the various sensors may be simultaneously pointed at different targets – a feature that will greatly increase the operational flexibility of the program.

Pointing commands for the following sensors are currently computed by the program:

(1) TRADEX
(2) ROTI telescope
(3) 48-inch telescope
(4) NIKE-X DR radar
(5) Optics sensors in PRESS aircraft

The process of computing the sensor pointing data from the track files is relatively simple and basically consists of a coordinate conversion from TRADEX rectangular coordinates to the pointing coordinates of the sensor. For some sensors, coordinate rate commands are also computed in addition to the pointing commands. These coordinate rates are applied to the sensor positioning servos as feed forward signals and have the effect of reducing errors due to servo lag.

In computing the pointing commands, the track file position coordinates are first translated from the origin of the TRADEX coordinate system to the location of the sensor. The sensor pointing coordinates are then computed from these translated rectangular coordinates. The fact that the track files are in rectangular coordinates greatly simplifies this translation process, which merely reduces to subtraction of the TRADEX coordinates of the sensor location from the target position coordinates in the track file. If the sensor in question is located more than one mile from TRADEX, the translated rectangular coordinates are also rotated to account for the curvature of the earth's surface. The elevation commands computed for pointing TRADEX and the ground optics sensors also include a correction for atmospheric refraction.

The above comments apply especially to the computation of pointing data for the airborne optics instruments that are carried aboard the PRESS aircraft tracked by the SKRs. In this case, the sensor location is the position of the moving aircraft given in the corresponding Aircraft Track File, and the pointing data are computed as follows. First the position and velocity components in the Aircraft Track File are subtracted from those in the Directing Track File. These translated target coordinates are then rotated from the TRADEX coordinate system to the orientation of a reference stable platform in the aircraft. The pointing commands are then computed from the target coordinates relative to this stable platform. The orientation of the aircraft platform is such that the platform remains parallel to the tangent plane at the earth location from which the aircraft was launched – normally Wake Island.

The equations used to perform the computations of the sensor command data described above and for the various sensors are given in the Appendix.

In addition to the pointing commands to the various PRESS sensors, the program also computes a 2-D prf command for TRADEX. 2-D (two-dimensional) prf computation is defined as the selection of a prf that not only keeps the target return out of the ambiguous (folded-over) range clutter but also out of the ambiguous doppler clutter. Because TRADEX tracks in both range and doppler, the selection of a prf that keeps the target return out of doppler clutter as well as range clutter greatly improves doppler tracking, particularly during re-entry where the doppler clutter becomes severe due to returns from the ionized wake of the re-entering target. The prf command is computed directly from the TRADEX range and doppler data sent to the computer, rather than from the data in the track files.

4.0 PLOTTING OF TARGET AND AIRCRAFT POSITION DATA

Target and aircraft positions are plotted by the program on two Milgo vertical plot boards that are located in the PRESS and aircraft control centers. These displays consist of 300- × 300-mile geographic overlay charts of the Kwajalein area on which the target and all aircraft positions are printed about once every ten seconds. The position data to be plotted are obtained directly from the associated track files. At present, the positions of all aircraft being tracked by SKRs, the target position in the Directing Track File, and the predicted re-entry position of the Directing Track File are plotted on these displays.

5.0 RE-ENTRY PREDICTION COMPUTATION

Predicted times and positions of re-entry are computed by the program for the ballistic trajectories in all existing TRADEX Target and Acquisition Track Files. Re-entry for the purposes of this computation is defined as the point at which the exoatmospheric ballistic trajectory of the target intersects a spherical surface located 300 kft above the earth's surface. The computation for each track file is performed by assuming that the ballistic trajectory of the target in the track file may be approximated by the Keplerian elliptical orbit that is defined by the current position and velocity components in that track file. The program then computes the intersection point of that orbit with the 300-kft sphere described above, and also determines the flight time along the orbit from the present target position to that re-entry intersection point.

The equations used to compute the predicted time and position of re-entry – based on the Keplerian elliptical target trajectory assumption described above – are given in the Appendix.

The predicted re-entry positions for the various track files thus obtained may then be plotted on the two Milgo plot boards described above. The predicted re-entry flight times for all track files are displayed on the on-line printer once every ten seconds and are also used to reset the external Time-to-Re-entry Countdown Clock when a manually executed TTR Clock reset command is given.

6.0 MISCELLANEOUS FUNCTIONS

In addition to data recording, generation of track files, computation of sensor directing data, and display generation for the Milgo plot boards, the program also performs the following miscellaneous functions:

(1) On-line typewriter I/O
(2) On-line teletype I/O
(3) Recording of anomalies on the Error Monitor Tape
(4) Generation of celestial pointing data

Most of these functions were mentioned briefly in the Introduction; a slightly more detailed description is given below.

6.1 ON-LINE TYPEWRITER

As mentioned earlier, the on-line typewriter is used for manual general-purpose data communication with the program during operations. The typewriter may be used either to query the contents of, or to enter data into, any computer storage location while the program is running. The subprograms that service the typewriter provide a great deal of flexibility in the format of the data which may be entered into, or displayed by, the typewriter. For example, the typewriter operator has the following choice of formats in which to communicate data through the typewriter: octal integer, decimal integer, and decimal (floating point). In addition, the operator can enter or display either single words or blocks of data in consecutive storage locations, the maximum block size being 16 words. References to storage may be made either by the octal address of the storage location or by the mnemonic symbol for that storage location. Because of its flexibility and ease of use, the typewriter is also a useful diagnostic tool for on-line debugging of program modifications or data link malfunctions.

6.2 ON-LINE TELETYPE CONNECTION

As mentioned earlier, the on-line teletype connection of the computer provides an I/O capability with either an external paper tape reader-punch or the external teletype lines to Kwajalein. Since the teletype connection is only used for special-purpose data communications, the formats of the teletype messages handled by the program are fixed. The teletype connection normally operates in the input mode, and the program provides special logic for detecting incoming teletype messages addressed to the PRESS computer. In the input mode, all messages on the teletype line enter the computer, but only those messages that are addressed to the PRESS computer are decoded and processed further by the program. The message address that is examined by the program consists of a key word in the heading of the message. The program also checks the contents of the message against the prescribed format and verifies the check sums of the data contained in the message. The program accepts the message for further processing only if verification of message format and check sums are satisfactory. For transmitting teletype messages from the computer, the teletype connection is put into the output mode by a manual command to the program. Upon completion of transmission of the message, the program automatically puts the teletype connection back into the input mode.

6.3 ERROR RECORDING

A means is provided for permanently recording all error conditions and other anomalies sensed by the program. Most of these conditions are recorded on the Error Monitor Tape for post mortem diagnosis, but conditions that require immediate operator attention are also displayed on the on-line printer. The recording of this information is handled by an Error Monitor Subroutine that can be called from any subprogram in which an error condition is sensed. The calling subprogram that senses the error transfers data identifying the error to the Error Monitor Subroutine for recording on the Error Monitor Tape. These data include the mnemonic name of the error and the contents of any computer storage locations associated with the error condition that will be useful in later diagnosis of the problem. This information is stacked in a buffer by the Error Monitor Routine for recording on the Error Monitor Tape and/or the on-line printer as soon as time is available.

6.4 GENERATION OF CELESTIAL POINTING DATA

The program has an option for generating directing data to point the optics sensors at celestial bodies, such as stars. This option is manually selected through a key on the computer console. The celestial coordinates (Sidereal Hour Angle and Declination) of the star, at which the optics sensors are to be pointed, are entered into the program either from the input parameter tape or through the on-line typewriter. The coordinates of the star are then computed in TRADEX rectangular coordinates, based on a near "infinite" radius of 10^9 ft, and these coordinates are placed into the Nominal Acquisition Track File. The equations used to perform the star pointing computations described above are given in the Appendix. The various sensors may then be pointed at the star by selecting the Nominal Track File as the Directing Track File, as described previously in Sec. 3.0.

7.0 SUBPROGRAM DESCRIPTIONS

As mentioned in Sec. 1.2, the RTP is really a collection of 29 semi-autonomous principal subprograms, each subprogram performing a unique function within the system. These subprograms can be considered as the building blocks of the RTP and enable the program to be organized in a modular manner. An outline of the various functions performed by the RTP has been given in Secs. 2 through 6, and some semblance of the order in which these functions are performed was implied by the context of those sections. Each of the functions described earlier is performed by separate subprograms or groups of subprograms in the RTP. Thus, before describing the over-all operation of the RTP as a whole – the order in which these functions are performed and how the subprograms are linked to each other – a brief description of each subprogram in the RTP will be given first. Table II lists the 29 subprograms that presently constitute the PRESS RTP. Omitted from this list are certain trivial table look-up programs, such as those used to perform refraction corrections, and all arithmetic function subroutines, such as those that compute the trigonometric and exponential functions – the latter being supplied as part of the Fortran Monitor Operating System for the 7094 Computer. The 29 principal subprograms can be broken up into the following seven functional groups:

(1) I/O buffering and data handling programs
(2) I/O device control programs
(3) Trajectory smoothing, extrapolation, and track file control programs
(4) Sensor directing data computation programs
(5) Re-entry prediction computation programs
(6) Plot computation programs
(7) Utility programs

A breakdown of the RTP subprograms listed in Table II into the above seven groups is given in Table III. Note that several of the subprograms listed in Table II appear in more than one group in Table III; this is because those subprograms perform several functions – each function being associated with a different group. A brief description of each group and of the subprograms included in it is given below.

7.1 GROUP I – I/O BUFFERING AND DATA HANDLING PROGRAMS

The general functions performed by the subprograms in this group are to convert, reformat, and buffer data exchanged between the program and the external I/O devices.

7.1.1 BUFFR

BUFFR is the principal I/O data buffering program in the RTP. One of its functions is to buffer and move data exchanged between the 7281 and the program. BUFFR also moves the data to be recorded on the output tape from program COMMON storage and from the 7281 input subchannel storage blocks to the computer output tape (RCT) buffer. BUFFR also decodes, converts, and applies corrections for refraction, etc., to the TRADEX input data samples that will be used to establish the TRADEX Target Track Files. Another function performed by BUFFR is to decode the settings of the external manual program controls entered from the computer console keys and from the remote control console at TRADEX. After decoding, BUFFR sets control flags in the program to reflect the settings of these external manual control switches.

TABLE II

RTP SUBPROGRAMS

(1)	ACDIR	Aircraft Directing Data Computation Program
(2)	ACONT	Data Channel "A" Control Program
(3)	ACSM	Aircraft Track File Smoothing Program
(4)	BCONT	Data Channel "B" Control Program
(5)	BETRAJ	Ballistic Trajectory Integration Program
(6)	BUFFR	7281 and Output Tape Data Buffering Program
(7)	DIDAT	TRADEX Directing Data Computation Program
(8)	DUMPR	Program Loader
(9)	ERROR	Error Monitor Program
(10)	FLTPG	Re-entry Prediction Control Program
(11)	INMES	Lift-Off Time and Acquisition Message Processor
(12)	INQUR	On-Line Typewriter Interpretation Program
(13)	KJMAIN	Main Program
(14)	NIKE-X	NIKE-X DR Directing Data Computation Program
(15)	OUTMES	ARIS Ship Acquisition Message Processor Program
(16)	PLATOS	48-inch Telescope Directing Data Program
(17)	PLOTC	Milgo Plot Control Program
(18)	PLOT	Milgo Plot Data Processor Program
(19)	PRFCON	TRADEX prf Computation Program
(20)	ROTI	ROTI Telescope Directing Data Program
(21)	SIM	Simulation Input Data Buffering Program
(22)	SIMCLK	PRESS Clock Simulator Program
(23)	STAR	Star Track Computation Program
(24)	TRAPC	Trap (Interrupt) Control Program
(25)	TTRPG	Re-entry Prediction Computation Program
(26)	TWXIOS	Teletype Data Processing and Control Program
(27)	TYOUT	Typewriter Output Data Processing Program
(28)	TYPEC	Typewriter Control Program
(29)	XTRAP	TRADEX Data Smoothing and Track File Control Program

TABLE III

BREAKDOWN OF RTP SUBPROGRAMS INTO FUNCTIONAL GROUPS

Group I – I/O Buffering and Data Handling Programs

(1) BUFFR
(2) ERROR
(3) INMES
(4) INQUR
(5) OUTMES
(6) SIM
(7) TWXIOS
(8) TYOUT

Group II – I/O Device Control Programs

(1) ACONT
(2) BCONT
(3) BUFFR
(4) TWXIOS
(5) TYPEC

Group III – Trajectory Smoothing, Extrapolation, and Track File Control Programs

(1) BETRAJ
(2) ACSM
(3) XTRAP
(4) STAR

Group IV – Sensor Directing Data Computation Programs

(1) ACDIR
(2) DIDAT
(3) NIKE-X
(4) PLATOS
(5) PRFCON
(6) ROTI

Group V – Re-entry Prediction Computation Programs

(1) FLTPG
(2) TTRPG

Group VI – Plot Computation Programs

(1) PLOTC
(2) PLOT

Group VII – Utility Programs

(1) DUMPR
(2) KJMAIN
(3) SIMCLK
(4) TRAPC

The centralization of these data handling functions in the BUFFR program enables the design of the other subprograms in the RTP to be freed from consideration of how and when the data communication with the external I/O devices is accomplished.

7.1.2 ERROR

The ERROR subprogram converts and buffers the error messages, generated by any subprogram detecting an error condition, for output to the Error Monitor Tape and/or the on-line printer. The ERROR program has the capacity for processing up to about 30 separate error messages per program cycle. All these messages may be recorded on the Error Monitor Tape, but the printing of messages on-line is limited by the speed of the on-line printer to about two error printouts per program cycle. After an error message has been processed, it is stored in an output buffer located in the ERROR subprogram until the requested output device (tape and/or printer) becomes available. The actual output operation is controlled by the BCONT or ACONT subprogram, as appropriate.

7.1.3 INMES

INMES is the Acquisition Track File message process or program. The INMES program decodes the target lift-off time entered through the on-line typewriter and adds it to the time-after-lift-off in the Nominal Track File initial conditions to obtain the real time of these initial conditions. The integration of the Nominal Trajectory is then started at the time given by this sum. INMES also decodes and verifies the contents of the offset trajectory message that enters the computer either through the on-line typewriter or through the on-line teletype connection. After this message is decoded, INMES transforms its contents into TRADEX rectangular coordinates and places the results into the Offset Track File.

7.1.4 INQUR

INQUR is the on-line typewriter input and inquiry interpreter program. Its function is to decode and convert all data entering the computer from the on-line typewriter. A message entered through the typewriter consists basically of a storage location address, data format specification, and data. The address may be given either in octal or as a mnemonic symbol. The input data are always numeric, but may be given in one of the following formats – octal-integer, decimal-integer, or decimal-number-with-decimal-point – as determined by the format specification given in the typewriter input message. The data format specification consists of: (1) the letter "O" immediately before the data for octal integer format, or (2) no specification for decimal integer format, or (3) the appearance of a decimal point in the data item for decimal-number-with-decimal-point format.

The INQUR program decodes the data according to the specified format and stores the converted data into the location(s) specified by the address portion of the message typed in. If the data portion of the typed-in message consists of a question mark, the message is interpreted as an inquiry request to display the contents of the location(s) specified in the address portion of the message on the typewriter. In this case, INQUR transfers to the TYOUT program, which formats the data reply requested for output to the typewriter.

7.1.5 OUTMES

The OUTMES program performs the inverse of the function performed by INMES, as the name might imply. The purpose of OUTMES is to format an acquisition message from a selected

track file for transmission to the ARIS ships via the on-line teletype connection. OUTMES encodes the position and velocity data from the selected track file and arranges these together with time-of-day into the specified teletype message format. OUTMES also computes and inserts check-sums for all data items in the message.

7.1.6 SIM

The SIM program is used to read previously recorded 7281 input data from an existing computer output tape (RCT), in this case referred to as the Simulation Input Tape, when the program operates in the Simulation Mode. The name SIM stems from the fact that the SIM program effectively simulates the operation of the 7281 input subchannels since these subchannels are turned off when the RTP operates in Simulation Mode. SIM reads the raw input data from the Simulation Input Tape into a buffer once a second. SIM then moves the data from this tape input buffer to the 7281 input subchannel storage blocks for pick-up by the BUFFR subprogram at the same time intervals as during regular Live Mode Operation.

7.1.7 TWXIOS

TWXIOS performs data encoding and decoding for communication between the program and the on-line teletype link. For output of data, TWXIOS first converts each character in the teletype output message generated by OUTMES from BCD to teletype code, and then moves the resultant encoded teletype message to the teletype subchannel output storage block in groups of eight characters. The converse function is performed for input data received from the on-line teletype connection. Each eight-character input block received is decoded from teletype code to BCD, and the decoded characters are blocked into groups of six BCD characters (one computer word). After conversion from teletype to BCD code, TWXIOS examines the incoming data and looks for a recognizable message heading indicating that the incoming message is addressed to the PRESS computer. No further processing on the input data is performed until a recognizable heading is encountered. All teletype data not associated with a recognizable message heading are discarded. However, when a recognizable heading indicating the presence of a message for the PRESS computer is encountered, TWXIOS signals the INMES program that an input message is being received and decoded, and is ready for further processing.

7.1.8 TYOUT

TYOUT has a function inverse to that performed by the INQUR program. The purpose of TYOUT is to assemble and format messages that are to be written out on the on-line typewriter. Specifically, TYOUT assembles the replies to inquiry requests made from the typewriter and interpreted by INQUR. The inquiry request interpreted by INQUR contains the address of the location(s) to be written out and the format according to which this reply is to be displayed.

7.2 GROUP II – I/O DEVICE CONTROL PROGRAMS

The general purpose of the subprograms in this category is to control the operation of the various I/O data channels connected directly to the computer. The external I/O devices connected to these data channels are then in turn controlled through the program generated commands transmitted to each channel. Basically these programs exercise control over the operation of the following three channels: 7607 Data Channel "A," 7607 Data Channel "B," and the

7281 Data Communication Channel. Data Channel "A" services the on-line printer and Milgo plotters; Channel "B," the various tape drives used by the program; and the 7281, as previously mentioned, consists of subchannels that are connected to the various external PRESS sensor data links. The 7281 also includes subchannels through which the on-line typewriter and teletype connection are attached to the computer. Each of the 7281 subchannels can be independently controlled by the program.

7.2.1 ACONT

As the name implies, ACONT is the control program for Data Channel "A." The I/O devices attached to that channel (the printer and plotters) are unique in that they are electromechanical in nature and consequently relatively slow. For example, the maximum data rate of the on-line printer is 150 lines of 72 characters a minute, which means that for each line printed, the printer ties up the data channel for about 400 msec or four program cycles. For the Milgo plotters, the situation is somewhat different. Although the actual transfer of data to the plotter only ties the data channel up for a fraction of a millisecond, the actual plotting cycle of the plotter is usually well over a second, depending on how far the plotting arm has to move from the previous point plotted. Thus, the maximum effective plotting rate is 60 points a minute or less.

Since the data channel can only service one I/O device at a time, the primary purpose of ACONT is to time-share the use of the channel between plotting and printing. Output requests for use of the printer and plotter are made to ACONT by various other subprograms in the RTP. ACONT examines these requests and allocates use of the channel for output to the requesting subprograms essentially on a first-come-first-served basis as time is available.

ACONT also monitors the actual time required for each output operation and disconnects the channel if the maximum allowable transfer time limit for the I/O device being used is exceeded. This is done because breach of the preset transfer time limits is indicative of an I/O device malfunction, and consequently this feature of ACONT prevents the channel and program from ever becoming "hung up" because of failure of an external I/O device.

7.2.2 BCONT

The function of BCONT is similar to that of ACONT, except that this subprogram controls the use of Data Channel "B." Because all the I/O devices attached to Data Channel "B" are high-speed tape units, whose data transfer time characteristics are uniform, the operation of BCONT becomes much simpler than that of ACONT. The data transfer time to any tape unit never exceeds two program cycles. BCONT time-shares the use of the data channel between the computer output tape, which is written out regularly once every second, the Error Monitor Tape, and when the program is in Simulation Mode, the Simulation Input Tape, which, when used, is read once every second. As in the case of ACONT, BCONT also prevents the channel and program from being "hung up" in the event of a tape unit malfunction by disconnecting the channel in the event that the maximum transfer time limit for a given tape unit is exceeded. BCONT also verifies each tape operation executed, and generates an error message to be displayed on the on-line printer in the event any redundancies (tape errors) are encountered when a tape is read or written.

7.2.3 BUFFR

The BUFFR program was briefly described previously under the Group I category of subprograms in Sec. 7.1.1. It is mentioned again here because, in addition to its data handling

functions it is also the control program for all 7281 subchannels, except those associated with the typewriter and teletype. The control functions performed by BUFFR in this regard essentially amount to keeping the operation of the 7281 subchannels synchronized with that of the program. This synchronization is accomplished by turning these subchannels on or off at appropriate times. In addition, BUFFR also monitors the operation of the subchannels for synchronization or data drop-out problems. In the event that a data transfer malfunction is detected, BUFFR causes appropriate error messages to be printed on the on-line printer and, when appropriate, also attempts to resynchronize the affected subchannel.

In addition to controlling and, to some extent, synchronizing the operation of the 7281 subchannels, BUFFR also effectively synchronizes the operation of the program with the time from the PRESS clock in the following manner: BUFFR is the first subprogram executed every program cycle. The RTP also returns to BUFFR at the completion of every program cycle. After return to BUFFR at the end of a program cycle, further execution of the RTP is delayed by a waiting loop in BUFFR until the next 0.1-sec mark from the PRESS clock. At that time, execution of BUFFR is resumed and a new program cycle is formally started.

BUFFR also controls the changes in program operation required when the program operates in the IF Playback or Simulation Mode. For example, when in Simulation Mode, BUFFR turns off all 7281 input subchannels since, during that mode of operation, the input data are read from the Simulation Input Tape by the SIM program. This function of BUFFR effectively isolates the operation of the remaining system subprograms from the effects of changes in program operating mode so that these subprograms may be designed without regard to what mode of program operation is being used.

7.2.4 TWXIOS

The TWXIOS program was also previously introduced in Sec. 7.1.7, under the Group I category of subprograms. It is discussed again in this section, since, in addition to the teletype data handling and conversion functions described in Sec. 7.1.7, TWXIOS also functions as the control program for the 7281 teletype subchannel. TWXIOS initially turns on the teletype subchannel in the input mode and the subchannel remains in this mode until a manual command is given to the program to transmit a teletype message. When the program is instructed to transmit a teletype message, TWXIOS turns the subchannel on in the output mode and transmits the specified message. As soon as transmission is completed, TWXIOS automatically puts the subchannel back into the input mode. When the subchannel is in the input mode, TWXIOS decodes and examines all incoming messages and looks for those addressed to the PRESS computer. When a message for the PRESS computer is encountered, it is saved for further processing by the INMES program. All other incoming messages are discarded.

7.2.5 TYPEC

TYPEC is the 7281 typewriter subchannel control program. The typewriter subchannel operates in any one of the three following modes: Inquiry, Input, and Output. The Inquiry Mode is a passive condition in which the typewriter can only signal the computer for attention. Except for the "Inquiry Request" Key (Attention), the typewriter keyboard is locked when the subchannel is in the Inquiry Mode. In the Input Mode, the typewriter keyboard is unlocked and all characters typed in on the keyboard enter the typewriter subchannel storage block in the computer. In the

Output Mode the keyboard is again locked, and the characters placed in the typewriter subchannel storage block by the program are typed out.

The purpose of the TYPEC program is to identify the cause of an interrupt initiated by the typewriter and to transfer to the appropriate subprogram (INQUR or TYOUT) for servicing that interrupt. TYPEC also turns the subchannel on in the appropriate mode whenever the typewriter mode of operation is to be changed.

As implied above, the subchannel normally remains in the Inquiry Mode waiting for an "Attention" signal (obtained by pushing the Inquiry Request Key) from the typewriter. Pushing the Inquiry Request Key when in the Inquiry Mode initiates a program interrupt that is recognized by TYPEC. TYPEC then turns the typewriter on in the Input Mode, the keyboard is unlocked, and the program becomes ready to accept input data from the typewriter. When typewriter interrupts occur while in the Input Mode, TYPEC transfers control to the INQUR program, which processes the incoming message. When the message is an Inquiry Request (a request for a display of data on the typewriter), the INQUR program transfers to TYOUT, which sets up the reply message (the data to be displayed) for output to the typewriter. TYOUT then transfers back to TYPEC, which turns the typewriter on in the Output Mode and the data set up by TYOUT are typed out. When typewriter interrups occur while in the Output Mode, TYPEC transfers control to TYOUT, which continues to set up the data for output to the typewriter until the output of all data requested has been completed. When a given input or output operation has been completed by the typewriter, TYPEC turns the subchannel back on in the Inquiry Mode, and remains passive until a new attention signal is received from the typewriter.

7.3 GROUP III – TRAJECTORY SMOOTHING, EXTRAPOLATION, AND TRACK FILE CONTROL PROGRAMS

The general purpose of the subprograms in this category is to establish and maintain track files (target trajectories) from which directing data for the various sensors may be computed. There are essentially three types of track files maintained by the program: TRADEX Target Track Files, Acquisition Track Files, and Aircraft Track Files. The TRADEX Target Track Files are smooth trajectories that are fitted in a least-squares sense to the target position measurements made by the TRADEX radar while tracking the target. The Acquisition Track Files are ballistic trajectories that are integrated in real-time from a set of initial conditions input to the program. The Aircraft Track Files, similar to the TRADEX Target Track Files, are smooth trajectories that are least-squares fitted to the aircraft position measurements made by the SKR aircraft tracking radars. The trajectory information in all these track files is given in TRADEX rectangular coordinates. Once a track file has been established, it is maintained by extrapolating or integrating its contents in real-time. The equations used for trajectory smoothing and integration of the track files are given in the Appendix.

7.3.1 BETRAJ

BETRAJ is the trajectory integration program that is used to extrapolate the TRADEX Target and Acquisition Track Files in real-time every program cycle. These track files are normally integrated by using the ballistic trajectory equations of motion for the acceleration components of the target in TRADEX rectangular coordinates. The total acceleration of a target in a ballistic trajectory – expressed in the TRADEX rectangular coordinate system – consists of the following

components: acceleration of gravity, apparent accelerations due to rotation of the earth-fixed TRADEX coordinate system in space as the earth rotates, and, during the re-entry portion of the trajectory, the effects of atmospheric drag.

However, for the TRADEX Target Track Files only, a special integration mode, consisting of simple straight-line extrapolation, is available for establishing track files on objects in non-ballistic trajectories. This special mode, named "Ballon Track Mode," may be manually selected from the computer or remote control consoles. It is principally used to maintain track files on balloon targets tracked by the radar during calibrations runs.

During the re-entry portion of a target trajectory, the acceleration components of the track file due to aerodynamic drag on the target are computed by BETRAJ in either of the following ways. For those track files, including the Acquisition Track Files, which correspond to targets not being tracked by TRADEX during re-entry, the target acceleration (or rather deceleration) component due to drag is computed by using a drag coefficient extracted from pre-stored tables of target drag coefficient vs Mach number or altitude. However, in the case of the track file associated with that target actually being tracked by TRADEX through re-entry, the drag deceleration components are derived directly from the target doppler and position measurements made by the radar. In either case, during re-entry the derived drag deceleration components are added to those due to gravity and earth rotation computed from the ballistic equations of motion. The total target acceleration, represented by this sum, is then used to integrate the trajectory in the associated track file.

7.3.2 ACSM

ACSM is the so-called "aircraft smoothing program." The function of this program is to establish and maintain the Aircraft Track Files. The Aircraft Track Files are established by recursively fitting a straight line track in a least-squares sense to the aircraft position measurements obtained from the SKR aircraft tracking radars and from the aircraft radar altimeter measurements transmitted to the computer via the A/G data link. A straight-line fit is used since the aircraft flight path is generally composed of straight-line segments, the flight time along each segment being on the order of several minutes. Consequently, the acceleration components of the Aircraft Track Files are set to zero. Each SKR is assigned to track a single aircraft. Furthermore, each of the three Aircraft Track Files is uniquely associated with a given SKR radar, and thus with the aircraft tracked by it. Consequently, the smoothed position and velocity of a given aircraft are always contained in the same Aircraft Track File.

Since the altimeter measurement made by the aircraft is a more accurate indication of the aircraft's vertical position than the SKR elevation measurement, the z-coordinate position and velocity in an Aircraft Track File are computed from the altimeter measurement, rather than from the SKR elevation measurement whenever the former is available. Consequently, the SKR elevation measurement is used only as a "back-up" in case of an altimeter or A/G data link malfunction. In any case, the ACSM program checks the values of the position measurements obtained from the SKRs and aircraft altimeters for consistency with the contents of the respective track files before using these measurements to update those track files. In this way, the Aircraft Track Files are guarded against corruption by spurious or extremely noisy measurements.

The ACSM program operates and samples the SKR and altimeter data for updating the Aircraft Track Files every program cycle. After a new data point has been "smoothed" or recursively fitted into a track file, the track file is extrapolated ahead in real-time by an increment

of 0.1 sec. When track on an aircraft is lost, the corresponding Aircraft Track File continues to be extrapolated in real-time until track is re-established by the SKR, at which time that track file is restarted from the new measurements.

7.3.3 XTRAP

The XTRAP program performs the recursive least-squares trajectory fitting (smoothing) of TRADEX track target position measurements into the TRADEX Target Track Files, and performs the control functions for both TRADEX Target and Acquisition Track Files. The track file smoothing process consists merely of recursively computing a least-squares fit trajectory to the radar tracking measurements of target position made by TRADEX. The target position measurements made by TRADEX are sampled for smoothing and updating the track files twice a second during the 0.1- and 0.6-sec program cycles. The data sample is used to update the corresponding track file only if TRADEX is actually tracking the target in both range and angle coordinates. The sampled coordinate measurements have already been corrected for refraction and propagation delay by the BUFFR subprogram. The trajectory fitted to the data is that integrated by the BETRAJ program, and is a ballistic trajectory unless the Balloon Track Mode of trajectory integration has been manually selected. Every track file, whether or not it is currently being updated with radar track measurements, continues to be extrapolated in real-time by the BETRAJ subprogram. The track file control functions performed by XTRAP will be described next.

For the Acquisition Track Files, track file control essentially consists of aligning the integration process with real-time. Both Acquisition Track Files are integrated along a ballistic trajectory from a set of initial conditions input to the program. However, the time tag of the given initial conditions may be earlier or later than the present value of real-time in the program. Consequently, the controls for the Acquisition Track Files speed up or delay, as necessary, the initial integration of these track files. For example, if the time of the initial condition for a given Acquisition Track File is later than the present value of real-time, commencement of integration for that track file by BETRAJ is delayed until the value of real-time from the PRESS Clock reaches the value of time given in the track file initial conditions. Conversely, if the time of the initial conditions for a given Acquisition Track File is earlier than the present value of real-time, that track file is integrated at a rate ten times real-time, until the time value of the track file contents catches up with the actual time from the PRESS Clock.

The track file control process for the TRADEX Target Track Files is closely associated with the trajectory fitting function and provides a means of automatically starting a new track file when TRADEX switches track to another target. The purpose of this track file control function is hopefully to uniquely associate each new target tracked by TRADEX with a different track file. The word "hopefully" is used above because, unfortunately, the operation of this control logic also sometimes causes a new track file to be started while TRADEX is still tracking the same target. This anomaly is especially likely to occur when the track of the given target by the radar is weak or unusually noisy. Briefly, the track file control logic for the TRADEX Track Files operates as follows: Each new TRADEX track data sample (the TRADEX track data is sampled for inclusion in the track files twice a second) is tested for consistency within certain limits in range and angle against the extrapolated contents of the most recent track file established from TRADEX data – the so-called Prime Track File. If the new data point falls within the given

limits, it is used by the recursive least-squares smoothing algorithm to update the Prime Track File. If it did not fit, it is used to start a new track file. This new track file is called the Alternate Track File. The new data point may fail the above consistency test with respect to the Prime Track File for one of two reasons: either the data point was spurious or excessively noisy, or it was obtained from a new track on another target. In the former case, the data point will effectively be discarded, thus guarding the existing Prime Track File against corruption by spurious data. In the latter case, a new track file on another target is considered to be formally established as soon as 20 mutually consistent data samples (10 seconds of data), none of which were consistent with the current Prime Track File, have been placed into the new Alternate Track File. When 20 mutually consistent data samples have been placed into the Alternate Track File, it then becomes the new Prime Track File, and the process continues. Note that each new data point is not tested for consistency with all existing target track files, but usually only with the current Prime Track File and then, if it did not fit the Prime, with the Alternate Track File, if one has been started. However, if the manually selected Directing Track File is not the same as the Prime Track File, the data samples are first tested for consistency with the Directing Track File before the test with the Prime Track File. Usually, however, the Prime and Directing Track Files will be one and the same. The preceding discussion of the TRADEX Target Track File control process has been somewhat abbreviated, and several important but specialized details have been omitted.

7.3.4 STAR

The STAR program is used to compute pointing data for directing the various PRESS sensors at a celestial body. This program operates only when "Star Track Mode" has been manually selected. The STAR program computes the position and "velocity" of the celestial body in TRADEX rectangular coordinates by assuming a fictitious range to the body of 10^9 ft; a value, which for all practical purposes, is equivalent to infinity. The position and velocity components of the body are computed in TRADEX rectangular coordinates for placement into a track file, because all the sensor pointing commands are computed from the rectangular coordinate position and velocity data in the track files by the sensor directing data computation programs. The velocity of a celestial body is an apparent one, due to the relative motion of the body with respect to the earth-fixed TRADEX rectangular coordinate system as that coordinate system rotates with the earth in space. The position and velocity coordinates of the celestial body or star are computed by STAR once each program cycle from the celestial coordinates (Declination and Sidereal Hour Angle) of the body that are entered into the program as input data on the Input Parameter Tape or through the on-line typewriter. The equations used to compute the position and velocity of the star, as described above, are given in the Appendix.

When the program operates in the Star Track Mode, normal ballistic integration of the Nominal Track File is suppressed, and the rectangular position and velocity coordinates of the celestial body computed by STAR are placed into that track file. Thus, to point the PRESS sensors at the star, the program is put into Star Track Mode and Nominal Track File is selected as the Directing Track File.

7.4 GROUP IV – SENSOR DIRECTING DATA COMPUTATION PROGRAMS

The general purpose of the subprograms in this category is to compute the directing or pointing data commands for the various PRESS sensors from the track files established and

maintained by the subprograms of Group III. At present, pointing commands for all PRESS sensors are computed from the same track file at any given time. The track file from which these commands are computed is called the Directing Track File. The Directing Track File may either be manually selected from the external program control consoles, or, if no manual selection of the Directing Track File is made, the program automatically selects the latest available TRADEX Target Track File (the Prime Track File) as the Directing Track File. Any of the following track files may be manually selected for computation of directing data: any one of eight TRADEX Target Track Files, Nominal or Offset Acquisition Track File, No. 3 Aircraft Track File, and Boresight Tower Track File. The Boresight Tower Track File, which has not been previously mentioned, merely contains the position coordinates of a fixed reference point, such as the location of the target simulator on the TRADEX boresight tower; this track file is normally used only for check-out and diagnostic purposes.

The functions performed by each of the Sensor Directing Data Computation Programs are all very similar. These subprograms operate during every program cycle and essentially merely convert the rectangular coordinate positions and velocities in the Directing Track File to sensor pointing commands and command rates in the pointing coordinate system for each sensor; normally, azimuth, elevation, and range, and their rates. The sequence of operations in this conversion process is normally as follows: translation of the track file position coordinates from the TRADEX origin to the sensor location, rotation of the translated position and velocity coordinates from the TRADEX system to the orientation of the corresponding rectangular system at the sensor location, to account for curvature of the earth in the event that the sensor is located more than a half mile from TRADEX, and, finally, conversion from sensor rectangular coordinates to pointing coordinates ($X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$ to $R, A, E, \dot{R}, \dot{A}, \dot{E}$). Elevation commands to TRADEX and the ground optics sensors are corrected for atmospheric refraction; the range command for TRADEX is also corrected for refraction. The computed commands for each sensor are also extrapolated from the track file data to correct for data transmission delays to each sensor, which vary from about 8 msec for ROTI pointing data to 150 msec for the aircraft optics pointing data. The equations used to perform the above computations of sensor pointing command data are given in the Appendix.

7.4.1 ACDIR

The ACDIR program computes commands for pointing the optics sensors located in the various PRESS aircraft being tracked by the SKRs. The directing data for each active aircraft are computed as follows: The target position and velocity coordinates in the Directing Track File are translated to the position and velocity of the aircraft by subtracting the contents of the corresponding Aircraft Track File from those of the Directing Track File. The translated coordinates are then rotated from the TRADEX coordinate system into a rectangular coordinate system aligned with the orientation of a stable platform in the aircraft. The orientation of the aircraft stable platform is that of the earth tangent plane at the launch site of the aircraft, usually Wake Island, and is known to the RTP a priori, the requisite rotation matrix having been entered into the program from the Input Parameter Tape. The rotated coordinates are then converted to look angle coordinates for the aircraft optics sensors relative to this stable platform. In addition to azimuth and elevation of the target from the aircraft relative to its stable platform,

the program also computes a slit roll angle command that is used by some of the optics sensors to align a viewing slit along the target's velocity vector.

7.4.2 DIDAT

DIDAT is the directing data computation program for the TRADEX radar. DIDAT computes the following commands for TRADEX from the data in the Directing Track File: range, azimuth, elevation, doppler (range rate), and azimuth, elevation, and doppler rates. Since the data in the Directing Track File are in TRADEX rectangular coordinates, the conversion to TRADEX look angle coordinates (range, azimuth, and elevation) is made directly from the track file data; no intermediate steps are required, as in the case of the other sensors. The elevation and range commands are corrected for atmospheric refraction. Although the above commands are continuously computed by DIDAT, the range, azimuth, elevation, and doppler commands are used by TRADEX only when the radar is not in full track and computer designation is specified at the TRADEX operating console. However, the computed rate commands are used as feed-forward signals to the TRADEX servo loops and may be used by TRADEX, even when the radar is in full track. The purpose of these rate commands is to reduce tracking and positioning errors due to servo dynamic lag.

DIDAT also assembles a program status message that is transmitted with the directing data but is used to drive indicator lights on the TRADEX remote RTP control console. The information in this status message includes such items as: (1) which track files are available for selection as Directing Track Files, (2) which track file is currently the Directing Track File, and (3) which track file is currently being "updated" with TRADEX track data.

7.4.3 NIKE-X

The NIKE-X program computes a target position data message from the data in the Directing Track File for transmission to the NIKE-X DR radar on Kwajalein. The computed message contains the target position in DR rectangular coordinates with parity check bits for each coordinate. The parity check bits are included to verify transmission reliability, since the data are transmitted to Kwajalein over a submarine telephone cable. The data in the message are computed as follows: First, the position coordinates in the Directing Track File are translated from the TRADEX origin to the DR location. Next, the translated coordinates are rotated from the TRADEX coordinate frame to that tangent to the earth's surface at the DR. The resultant target position in DR rectangular coordinates is then placed in the message for transmission.

7.4.4 PLATOS

The PLATOS program computes the pointing commands to be transmitted to the 48-inch telescope. PLATOS computes the following commands from the data in the Directing Track File: azimuth, elevation, and focus. The focus command is a function of the range to the target computed from the track file position coordinates. The azimuth and elevation commands are computed in the same manner as for all other sensors that are not located at the TRADEX origin: translation of coordinates from TRADEX to the location of the 48-inch telescope, rotation of coordinates to account for the earth's curvature between the locations of TRADEX and the telescope, and conversion from rectangular coordinates at the telescope to azimuth and elevation. Finally, the computed elevation command is corrected for refraction of light by the atmosphere.

7.4.5 PRFCON

The PRFCON program computes a pulse repetition frequency (prf) command for the TRADEX radar. Unlike the other directing data computation programs, PRFCON computes its command directly from the TRADEX range and doppler data decoded by BUFFR, rather than from the data in the Directing Track File. The prf command computed by PRFCON is defined as being either 2-D (two-dimensional) or 1-D, according to whether the computed prf is such that the ambiguous target return will be kept out of both the ambiguous range and doppler clutter or only out of the ambiguous range clutter, respectively. Obviously, every prf command that satisfies the 2-D criteria also satisfies the 1-D criterion, but the converse is not necessarily true. The prf command computed by the program is restricted to a set of twelve discrete values of prf at which TRADEX can operate. The values of these twelve prf's are contained in a table built into the program. The program attempts to select a prf command from the available set that will satisfy the 2-D criteria – that is, will keep the ambiguous target return out of both the ambiguous range and doppler clutter. Depending on the current values of TRADEX range and doppler, and on the widths of the clutter regions, it may happen that several of the available values of prf satisfy the 2-D criteria, or that none do. The widths of the range and doppler clutter regions are measured by TRADEX before a test operation and are input to the program either from the Input Parameter Tape or the on-line typewriter. When several values of prf satisfy the 2-D criteria, the program chooses that value that will continue to keep the target out of both clutters for the longest period of time. In the latter case, where no prf's satisfying the 2-D criteria can be found in the available set, the program then selects that prf satisfying the 1-D criterion that will continue to keep the target return out of the range clutter for the longest period of time. The available set of prf's is such that at least one prf satisfying the 1-D criterion can almost always be found. This abbreviated description of the function and operation of the PRFCON program is intended merely to convey a general idea of the prf selection process performed by that program. The actual details of the prf selection logic in the program are quite complex and have been omitted in the above description.

7.4.6 ROT1

The ROT1 program computes the pointing commands for the ROT1 telescope from the data in the Directing Track File. The functions performed by, and operation of, the ROT1 program are nearly identical to those of the PLATOS program described in Sec. 7.4.4, with the exception that no focus command is computed for the ROT1 telescope.

7.5 GROUP V – RE-ENTRY PREDICTION COMPUTATION PROGRAMS

The purpose of the subprograms in this category is to compute the predicted time and position of re-entry for each available TRADEX Target and Acquisition Track File maintaining a ballistic trajectory on a target that will re-enter the atmosphere. Naturally, no re-entry predictions are made for track files maintaining trajectories on satellites, balloon targets, star tracks, or ballistic trajectories that have already re-entered the atmosphere. The re-entry predictions are based on the assumption that the ballistic trajectory of a target, from its present position in a track file to the altitude of re-entry, may be approximated by a Keplerian orbital ellipse. The orbital elements of this ellipse are determined from the present values of position and velocity in that track file. The altitude of re-entry to which the predictions are made is

300 kft. The predicted times of re-entry for each track file are used to generate a display of differential flight times to re-entry for each track file relative to the Nominal Track File – this display being printed out on the on-line printer about once every ten seconds. In addition, the predicted time of re-entry for a given selected track file may be used to reset the external time-to-re-entry (TTR) clock. The track file designated for this purpose is manually selected from the computer console keyboard. The predicted re-entry positions of all track files for which this computation is made are available for display on the off-line Milgo plotters. The re-entry prediction computations and associated display are only performed about once every ten seconds for each track file since, for valid ballistic trajectories, the predicted values change very slowly, if at all. Consequently, the consistency of the prediction data displayed also provides a check on the validity and consistency of each corresponding track file. The equations used in performing the re-entry prediction computations described above are given in the Appendix.

7.5.1 FLTPG

FLTPG is the control program for performing the re-entry prediction computation. This program only operates once a second, during the 0.4-sec program cycle and sequences the computations for all track files so that a different track file is processed every second. FLTPG also examines the altitude of each track file being processed and causes the prediction computations to be performed only for those track files whose altitudes are in excess of 400 kft. The actual prediction computations are performed by TTRPG, which is discussed in Sec. 7.5.2.

FLTPG also generates the differential flight time display for the on-line printer. This display consists of the predicted lift-off to re-entry flight time for the Nominal trajectory and the differential flight times for all other track files relative to the Nominal. This display provides a means of determining whether the target in a given track file is ahead of, or behind, the Nominal. FLTPG also contains the logic for resetting the external TTR clock from the predicted re-entry time of the track file manually selected for this purpose.

7.5.2 TTRPG

TTRPG performs the actual re-entry prediction computations for each track file. This program is called by FLTPG whenever the re-entry prediction computations for a given track file are to be performed. TTRPG determines the Keplerian orbital ellipse approximating the track file ballistic trajectory from the current values of the position and velocity in the track file. TTRPG then determines the intersection point of this ellipse with a sphere 300 kft above the earth at TRADEX, and computes the resultant position of the intersection point and the remaining time-of-flight along the elliptical orbit to that point. In the event that the track file contains a satellite trajectory, no intersection point of the orbital ellipse with the 300-kft sphere will exist and the remaining computations in TTRPG are skipped. In that case, the word "ORBIT" will be printed in lieu of the differential flight time for that track file in the differential flight time display on the on-line printer.

7.6 GROUP VI – PLOT COMPUTATION PROGRAMS

The purpose of the plot computation programs is to generate a geographic display plot of the aircraft positions in all active Aircraft Track Files, the current target position in the selected Directing Track File, and the predicted re-entry position of the target in the Directing Track

File. These positions will be plotted on two Milgo vertical display plotters on a 300- × 300-mile geographic overlay chart of the Kwajalein area. The actual physical size of each display is about 3 feet square. The function of the plot computation programs is to scale the positions to be plotted to the scale of the plot boards, and to reformat the position data from the track files into commands for positioning the plotting arms of the plot boards. Since the Milgo plot boards can only plot one point at a time, at a maximum rate of about one point a second, these programs also control the times at which the individual points are plotted, thus effectively time-sharing the use of the plot boards.

7.6.1 PLOTC

PLOTC is the plot control program. The function of this program is to sequence the plotting of all position points to be displayed, and to set up the data defining each point to be plotted for scaling and reformatting by the PLOT program. PLOTC is executed only four times a second during 0.6-, 0.7-, 0.8-, and 0.9-sec program cycles and only if plotting has been manually enabled by a command from the computer console. On each entry, PLOTC sequences the processing of each point to be plotted, so that a different point will be processed for plotting on each consecutive entry.

A total of twelve different plot points are processed – six for each of the two Milgo plot boards. Presently, the displays on each of the two plot boards are identical, so that six identical pairs of plotting operations are performed by PLOTC. However, the processing of each of the six plotting operations is performed separately by PLOTC for each of the two plot boards – for a total of 12 operations. The processing performed by PLOTC consists of determining whether or not the current data point is available for plotting (whether or not the associated track file is available), and, if available, setting up the necessary plot data for scaling and reformatting by the PLOT program.

Although PLOTC operates four times a second, the timing characteristics of the Milgo plot boards restrict the number of points that may actually be plotted to a maximum of one point per second on each board. Therefore, the number of points processed by PLOTC that are actually to be plotted is effectively restricted to a maximum of two points each second – one plot point for each of the two boards. The only circumstances under which PLOTC can actually perform four separate plot processing operations each second – a new one on each entry – is when at least two of the four operations being processed do not result in a point to be plotted. When every plot processing operation performed by PLOTC results in a point to be plotted, the effective processing rate is reduced to two points a second, and six seconds are required to complete the entire 12-operation plotting cycle.

A new plotting cycle is initiated by PLOTC once every ten seconds on the 9.6-sec program cycle. Because a maximum of six seconds is required to complete the plotting operations in every cycle and a new cycle is only initiated every ten seconds, an unobstructed view of each plot board is obtained during the last four seconds of each plotting cycle, or 40 percent of the time.

The six plotting operations performed by PLOTC for each of the two plot boards is given below.

Plotting Operation	Plotting Symbol
(1) Plot Directing Track File Position	T
(2) Plot Aircraft No. 1 Track File Position (if active)	0
(3) Plot Aircraft No. 2 Track File Position (if active)	+
(4) Plot Aircraft No. 3 Track File Position (if active)	×
(5) Plot Predicted Re-entry Position of Directing Track File	TF Nos. 1 through 8, N, 0
(6) Return Plotter Arm to Side of Plot Board	–

The last step is performed to provide an unobstructed view of the plot boards until the next plotting cycle is begun.

7.6.2 PLOT

PLOT is called by PLOTC whenever a plot data point set up by PLOTC is to be further processed for actual plotting. The PLOT program performs the plot scaling computations and the formatting of the position and plotting symbol commands to be sent to the Milgo plotters. A different plotting symbol is used for each data item plotted. PLOT also discards all points whose position falls outside the range of the plot boards. After scaling and formatting of the commands, PLOT sets up a plotter-select request to the ACONT program to transmit the plot data to the designated Milgo plot board.

7.7 GROUP VII – UTILITY PROGRAMS

The programs in this category are not related to any of the principal operational functions performed by the RTP. They do, however, perform certain support or utility functions essential to the operation of the RTP, per se in much the same way that the utilities of gas, water, and electricity are essential to the operation of a home or commercial building. Also, unlike the subprograms in the previously described categories, the operations performed by the individual utility subprograms are relatively unrelated – in much the same way that the uses of gas, water, and electricity are all unrelated – though essential. Finally, the functions performed by these programs affect only the internal operation of the RTP – in effect, they perform certain housekeeping operations within the RTP. Thus, their operation remains unnoticed by, and has little interest for, the observer who is primarily interested in the external manifestations of the functions performed by the RTP. A summary analog of the role of these subprograms in the RTP, for the more technically inclined reader, would be the relationship of power supplies and timing generators to the operation of a radar such as TRADEX.

The specific utility programs to be described perform functions associated with loading the program into the computer from a magnetic tape, reading the Input Parameter Tape, initial start-up of the program, sequencing execution of all the subprograms every program cycle, servicing of program interrupts initiated by the 7281 subchannels and other devices, and providing a synthetic means of program synchronization during IFPB operations.

7.7.1 DUMPR

The function of the DUMPR program is to load the RTP into the computer from the Real-Time System Tape (a magnetic tape containing a library of programs associated with and including

the RTP) and to read the Input Parameter Tape. When the RTP has been loaded and the contents of the Input Parameter Tape have been read into the program, DUMPR transfers to KJMAIN, the next program to be discussed, and operation of the RTP is initiated. As should be expected, DUMPR is executed only once, during the initial start-up of the RTP.

7.7.2 KJMAIN

KJMAIN is the "main program" of the RTP. It is the main program only in the sense that its primary function is to call the principal subprograms of the RTP and get them executed in sequence every program cycle. In this capacity, KJMAIN also logs the actual execution times required by each subprogram every program cycle.

KJMAIN also performs the following subsidiary functions: KJMAIN writes the so-called "calibration record" as the first physical record on the computer output tape. This calibration record contains input data read from the Input Parameter Tape; it is recorded on the output tape to provide a permanent record of these data. KJMAIN also provides a "program restart" facility by which the program can be restarted without loss of existing track files in the event of an unexpected and temporary program or computer failure. The Floating Point Trap Error Monitor Routine is also located in the KJMAIN program. This routine processes the conditions associated with the occurrence of a floating-point trap (invalid floating-point arithmetic operation) for recording on the Error Monitor Output Tape. Finally, KJMAIN performs the functions required for orderly shut-down of the program when the manual "end program" command is entered on the computer console. These functions consist of end-of-filing and rewinding all tapes used by the program and turning off the 7281.

7.7.3 SIMCLK

SIMCLK is the PRESS Clock Simulator Program. This subprogram operates only when the RTP is in the IFPB Mode. Its function is to generate a program pseudo-clock word from the 100-pps timing signals recorded on the IF tape that are sent to the computer through the TRADEX Gain Input Subchannel. The arrival of each timing pulse from the IF tape causes the above subchannel to interrupt the program, thus activating SIMCLK which then increments the pseudo-clock by 0.01 sec. The format of this program-generated clock word is identical to that received by the program from the PRESS Clock during Live Mode operations. This pseudo-clock word generated by SIMCLK plays the role of the PRESS Clock when the program operates in the IFPB Mode and is used to synchronize operation of the program during IFPB operations.

7.7.4 TRAPC

The function of the TRAPC program is to identify the source of program interrupts initiated by the various 7281 subchannels that are enabled to interrupt the computer. When these interrupts occur, TRAPC transfers to and executes the appropriate interrupt servicing routine for each subchannel interrupt requiring service. Presently, the 7281 subchannels that generate interrupts requiring service by the program are: TRADEX and Ground Optics Output, TRADEX Gain Input, Typewriter, and Teletype. The interrupt servicing routines for the first three subchannels above are located in the BUFFR subprogram. The Typewriter Interrupt servicing routine consists of a single instruction in TRAPC, and the Teletype Interrupt servicing routine is in the TWXIOS subprogram.

8.0 PROGRAM OPERATION

Section 1, the Introduction, briefly outlined the role of the RTP in controlling the various PRESS sensor systems and described the general structure of the RTP and its relationship to the external I/O devices with which it communicates. Following this outline, the material in Secs. 2 through 6 provided a somewhat more detailed examination of the internal functional operations performed by the RTP. Sections 2 and 3 described the concept and purpose of the various track files generated by the program and their role in the generation of sensor directing data. Sections 4 and 5 discussed the subsidiary operations of the display on Milgo plotting boards of the target and aircraft positions in the track files, and the re-entry prediction computations for targets having ballistic trajectories. Finally, Sec. 6 covered the various miscellaneous internal functions performed by the program – communication with the on-line typewriter and the on-line teletype connection, recording of error conditions and other operational anomalies, and generation of celestial pointing (Star Track) data for the optics sensors.

Following this over-all description of the functions performed by the RTP, a brief outline of each of the individual subprograms in the RTP, which actually perform these functions, was given in Sec. 7.

Having described in Sec. 7 the function and operation of the individual subprograms, which together constitute the RTP, we shall now examine the operation of the RTP as a whole. In other words, we shall examine how the building blocks of the RTP – its subprograms – fit and function together. The over-all operation of the RTP can be examined from either one of two points of view – the operational point of view from the standpoint of the chronological flow of execution for the individual subprograms, or the functional point of view from the standpoint of data flow through the program. The operational point of view will be examined first.

The execution flow diagram for the RTP is shown in Fig. 3. A simple program cycle consists of one complete circuit around this diagram, starting with the BUFFR subprogram. Note that the KJMAIN program appears only once in the diagram, immediately before the BUFFR subprogram. However, since all the subprograms are called by KJMAIN in its role as the main program, the transfer from one major subprogram to the next is actually accomplished through a return to KJMAIN, as indicated by the asterisks on the diagram.

As seen from the diagram, the DUMPR program initially loads the program into the computer and reads the Input Parameter Tape. DUMPR then transfers to KJMAIN, which writes the "calibration" record on the output tape and performs certain internal initialization functions required before actual execution of the first program cycle is begun. KJMAIN then transfers to BUFFR to begin the first program cycle.

8.1 OPERATION OF A TYPICAL PROGRAM CYCLE – A GUIDED TOUR

The operation of a "typical" program cycle will be described in this section. A typical program cycle is any program cycle other than the first cycle with the program operating in the Live Mode and with Star Track Mode not enabled. The execution of the first program cycle is somewhat different from a typical one. In the first program cycle, various initialization functions, such as the initial turn-on of the 7281 subchannels, are performed by the BUFFR subprogram, and the execution of the BCONT subprogram is skipped because there are no data yet to be written out. The operation of a program cycle in Simulation Mode differs from a typical one in Live Mode only in the following: the operation of the BUFFR subprogram is slightly different in Simulation

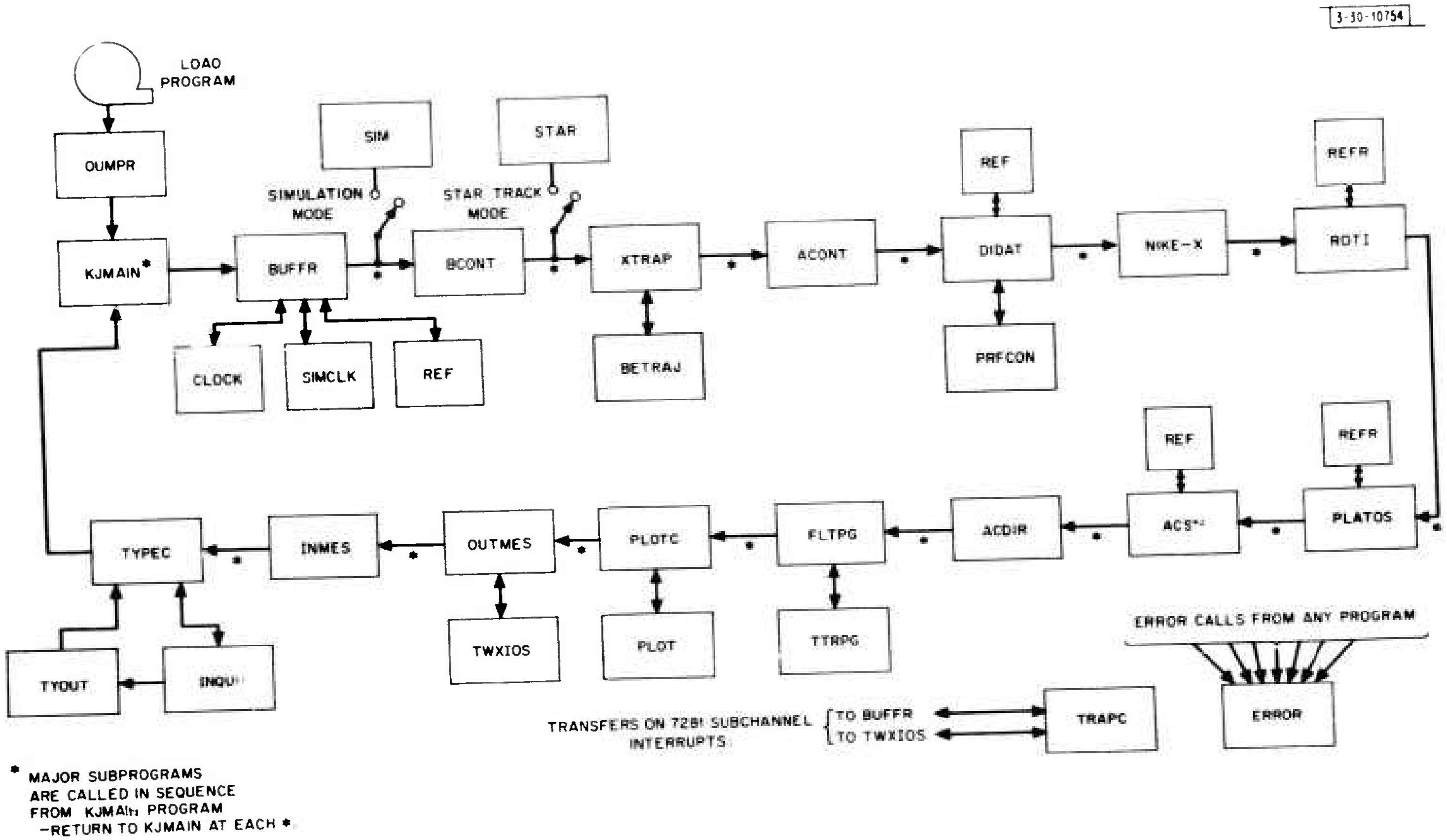

Fig. 3. RTP execution flow diagram.

Mode than in Live Mode, the SIM subprogram is executed immediately after BUFFR and before BCONT, and the BCONT subprogram causes the Simulation Input Tape to be read once a second during the 0.8-sec program cycle. In IFPB Mode, the only difference in the operation of a program cycle, in contrast to Live Mode, is in the BUFFR subprogram. The BUFFR subprogram operates somewhat differently in IFPB mode because it synchronizes the program to the 100-pps timing signals from the IF tape rather than to the PRESS Clock as in Live Mode. We now return to the description of the operation of a typical program during Live Mode operation.

8.1.1 BUFFR Operation

At the start of a typical program cycle, program execution is "hung up" in a waiting loop in BUFFR waiting for T00-msec time from the clock. The T in T00 msec is the number of the integral 0.1 sec coming up, i.e., 0, 1, 2, 3. . . 9. At T00 msec, program execution is released from this waiting loop, and a new program cycle is formally begun. BUFFR then moves the TRADEX input data that arrived during the previous program cycle from the 7281 TRADEX input subchannel storage blocks to the output tape buffer. BUFFR also decodes and converts the most recent TRADEX data sample (that at the 0.X9 sec, where X = T − 1) and performs a set of acceptance tests on the data to determine if TRADEX is in full track and if the data are suitable for inclusion in the track files. During this process, BUFFR also corrects the TRADEX elevation and range data for refraction and the range data for radar propagation delay. The converted and corrected TRADEX data are placed into the program COMMON storage area for later use by the XTRAP and PRFCON subprograms. In addition to the above functions, during the zero-tenth second program cycle only, BUFFR also resynchronizes the operation of the TRADEX input subchannels, decodes the external manual control data entered from the computer console and/or the TRADEX remote computer control console, and moves the program status words, collectively known as the "Flag Block" to the output tape buffer for recording. Finally, during the 0.2-sec program cycle only, BUFFR also moves the contents of all track files to the output tape buffer for recording.

8.1.2 BCONT Operation

As seen from Fig. 3, during a "typical" program cycle, the BCONT subprogram is executed immediately following the execution of BUFFR. On the zero-tenth second, BCONT initiates the writing of the contents of the output tape buffer – set up by the BUFFR program – onto the computer output tape. This output operation is completed within two program cycles. On any program cycle after the 0.1 sec, BCONT writes the contents of the Error Monitor Buffer onto the Error Monitor Output Tape only if any error data have been previously placed in this buffer by the ERROR program. An Error Monitor Tape write-out is completed within one program cycle. When the program operates in the Simulation Mode, BCONT also initiates the read-in of input data for the next second from the Simulation Input Tape on the 0.8-sec program cycle. This read-in operation is also completed within two program cycles.

8.1.3 XTRAP and BETRAJ Operation

The XTRAP subprogram is executed following BCONT. The XTRAP program calls BETRAJ and gets all existing TRADEX target and acquisition track files integrated ahead by 0.1 sec every program cycle, thus maintaining the trajectories in these track files in real-time. These track

files are stored in the program COMMON storage area for later recording and use by the sensor-directing data computation programs, the plot programs, and the re-entry prediction programs. XTRAP also contains the control logic for synchronizing the initial integration of the acquisition track files. During the 0.1- and 0.6-sec program cycles, before the track files are integrated ahead, XTRAP also samples the TRADEX Track data for updating the current Prime TRADEX Target Track File or starting a new (Alternate) TRADEX Target Track File. The TRADEX data sample taken at this time is used for updating the appropriate track file only if TRADEX is in full track mode and the TRADEX AGC voltage exceeds a minimum threshold. These conditions are checked by the BUFFR program during the process of TRADEX data decoding. If the above conditions are met, the data point is tested for consistency with the Prime Track File, and, if not consistent with it, is then tested for consistency with the Alternate Track File. If the data point is also inconsistent with the Alternate Track File, it is used to restart that track file. The data point is then used to update the track file with which it was consistent; the updating process consists of the recursive least-squares smoothing algorithm.

Finally, the XTRAP program also places the selected Directing Track File in a separate COMMON storage area block for use by the sensor-directing data programs. If no manual selection of the Directing Track File is made, XTRAP automatically selects the Prime Track File as the Directing Track File.

8.1.4 ACONT Operation

The ACONT subprogram is executed following execution of XTRAP and BETRAJ. ACONT searches for requests for output to the printer or Milgo plotters that have previously been made by the other subprograms: principally, FLTPG, ERROR, and PLOT. When such a request is encountered, ACONT initiates the desired output operation if Data Channel "A" is free.

8.1.5 Operation of Sensor Directing Data Computation Subprograms – DIDAT, PRFCON, NIKE-X, ROTI, and PLATOS

Following the execution of ACONT, the above sensor directing data computation programs are executed in the order shown in Fig. 3.

Except for PRFCON, all these programs compute the directing command data for their respective sensors from the contents of Directing Track File stored in COMMON by XTRAP. The computed directing data are stored in program COMMON storage in two formats. The directing data are stored in actual command format (fixed point) for transmission to the sensors through the 7281. The directing data are also stored in floating-point format for recording on the output tape. These data will be moved from COMMON to the 7281 output subchannel storage areas and to the output tape buffer by the BUFFR subprogram immediately before the start of the next program cycle.

PRFCON, which is called by DIDAT, computes the TRADEX prf command directly from the TRADEX range and doppler sample previously decoded by BUFFR. The prf command is also stored in COMMON in two formats, along with the other command data for TRADEX.

DIDAT also prepares a program status message for display on the TRADEX remote control panel. This message is also stored in COMMON, along with the other TRADEX command data.

8.1.6 Operation of Aircraft Subprograms ACSM and ACDIR

The aircraft subprograms ACSM and ACDIR are executed next, in the order shown. ACSM generates the Aircraft Track Files. ACSM samples the SKR and aircraft altimeter data stored

in the 7281 SKR input subchannel storage block, and uses these data to update the Aircraft Track Files which are also stored in program COMMON storage. The updating process again consists of a recursive least-squares smoothing algorithm. The Aircraft Track Files are then extrapolated ahead by 0.1 sec, thus maintaining the aircraft tracks in real-time.

The airborne optics directing data computation program, ACDIR, is executed immediately after ACSM. The ACDIR program computes the target position relative to each of the various aircraft by subtracting the contents of each Aircraft Track File from those of the Directing Track File. ACDIR then computes the pointing commands for all existing aircraft (up to three) from these relative positions and velocities. The resulting aircraft pointing commands are stored in program COMMON storage in the same manner as the pointing commands previously computed by the other sensor directing data computation subprograms, and will later similarly be transmitted to the 7281 output subchannels and the output tape buffer by the BUFFR subprogram.

8.1.7 Operation of Re-entry Prediction Subprograms – FLTPG and TTRPG

FLTPG is the next subprogram executed after ACDIR, but it is only executed during the 0.4- and 0.8-sec program cycles. During all other program cycles, execution of this program is skipped.

On its execution during the 0.4-sec program cycle, FLTPG processes a TRADEX Target or Acquisition Track File located in program COMMON storage in preparation for the re-entry prediction computation. A different track file is processed each second. If the target altitude in the track file being processed is above 400 kft, FLTPG calls TTRPG to perform the actual re-entry prediction computations for that track file; otherwise, further processing of that track file is skipped. The predicted re-entry positions computed by TTRPG for each track file are placed in program COMMON storage; the predicted flight time to re-entry is returned to FLTPG. After all ten TRADEX Target and Acquisition Track Files have been processed once every ten seconds, FLTPG also sets up a select request to the ACONT program to print the differential flight-time-to-re-entry display on the on-line printer.

On its execution during the 0.8-sec program cycle, FLTPG performs the logic for resetting the external TTR clock to the predicted re-entry time of a selected track file, if a manual selection of that track file has been made. The TTR clock can be reset from the program at two discrete values of time: two minutes and eight minutes to re-entry.

8.1.8 Operation of Plotting Subprograms PLOTC and PLOT

The next subprogram executed following FLTPG is PLOTC, the plot control program. PLOTC is only executed four times a second during 0.6-, 0.7-, 0.8-, and 0.9-sec program cycles, and only if plotting has been manually enabled from the computer console; during all other program cycles, execution of this program is skipped. A different plotting operation (aircraft position plot, Directing Track File position plot, etc. separately for each of the two Milgo plot boards) is processed by PLOTC on each consecutive entry. The data to be plotted are taken from the track files and the predicted track file re-entry position data in program COMMON storage. If the plot point being processed exists, PLOTC calls the PLOT program to scale the position of the point to the plot board scale, and to format the positioning and plotting symbol commands for transmission to the specified plot board. The PLOT program then also sets up a select request to the ACONT program for transmitting the plotting command to the specified plot board via Data Channel "A."

8.1.9 Operation of On-Line Teletype and Acquisition Message Processing Subprograms – OUTMES, TWXIOS, and INMES.

Following PLOTC, the OUTMES subprogram is executed every program cycle. OUTMES is the teletype output acquisition message processing program. OUTMES also always calls TWXIOS, the teletype processing and control program and, if nothing else is going on, makes certain that the 7281 teletype subchannel is turned on in the input mode. However, if a manual control command has been executed to transmit an acquisition message to the ARIS ships via teletype, OUTMES prepares this message in a specified format from the Directing Track File data located in program COMMON storage. OUTMES then calls TWXIOS, causing that program to turn the 7281 teletype subchannel on in the output mode. TWXIOS then automatically initiates transmission of this message through the teletype subchannel. The actual transmission of the message takes several program cycles. TWXIOS independently completes transmission of the message (under control of teletype subchannel interrupts) while the subsequent program cycles are being executed. On a subsequent program cycle after the transmission of the message has been completed, OUTMES, in its regular call of TWXIOS, causes the teletype subchannel to be turned back on in the Input Mode. When the teletype subchannel is in the Input Mode, TWXIOS automatically accepts and decodes any messages received from the external teletype connection that are addressed to the PRESS computer. This is again done independently under control of the teletype subchannel interrupts while the rest of the program is operating regularly.

Following the execution of OUTMES, the INMES subprogram is executed once a second during the 0.4-sec program cycle. During all other program cycles, execution of INMES is skipped. If a lift-off time has entered into the program during the previous second, INMES decodes this item, converts it to seconds and adds it to the time-after-lift-off in the Nominal Track File initial conditions to obtain the real-time of the initial conditions for that track file. INMES also processes any acquisition message received from the on-line teletype connection through the TWXIOS subprogram, converts the data in that message to TRADEX rectangular coordinates, and places the result into the Offset Track File. Both the Nominal and Offset Acquisition Track Files are located in program COMMON together with the other track files. If neither a lift-off time nor an acquisition message was entered into the program during the previous second, operation of the INMES program is effectively skipped.

8.1.10 Operation of Typewriter Subprograms – TYPEC, INQUR and TYOUT

Finally, following INMES, the TYPEC subprogram is executed every program cycle. TYPEC is the control program for the on-line typewriter; it initially turns the 7281 typewriter subchannel on in the Inquiry Mode and, subsequently, checks to see if an interrupt from the typewriter has occurred during the past program cycle. If an interrupt did occur, TYPEC determines whether the subchannel was in the Inquiry, Input, or Output Mode when the interrupt was initiated, and calls either the INQUR or TYOUT subprograms, as appropriate, for processing the data communication with the typewriter. An interrupt in the Inquiry Mode is an attention signal to the program, and in this event, the only action taken in TYPEC is to turn the typewriter subchannel on in the Input Mode, thus unlocking the typewriter keyboard for input. If the interrupt occurred in the Input Mode, TYPEC calls INQUR, which decodes and processes the message being typed in. If the interrupt occurred in the Output Mode, TYPEC calls TYOUT, which processes and encodes data for output to the typewriter. TYPEC also controls the mode of operation of the typewriter

subchannel – switching the subchannel from the Inquiry Mode to Input or Output Mode, or back to Inquiry Mode, as required. TYPEC is effectively the last subprogram executed every program cycle.

8.1.11 Return to BUFFR to Prepare for Next Program Cycle

Following the execution of TYPEC, program execution again returns to the BUFFR subprogram via KJMAIN. The total time required in each program cycle is always less than 98 msec. so that this return to BUFFR will always be made before the next program cycle is scheduled to commence. On return to BUFFR, the program "hangs up" in a waiting loop until 98-msec time – 2 msec before the next program cycle is scheduled to begin. When the program is released from this waiting loop, BUFFR moves the sensor directing data commands previously computed by the sensor directing data computation subprograms from program COMMON storage to the 7281 output subchannel storage blocks and distribution buffers, and the corresponding command data in floating point format, for recording, from COMMON to the output tape buffer. During this time interval, BUFFR also moves the input data from the SKR, A/G, and Ground Optics Readback 7281 input-subchannel storage blocks to the output tape buffer for recording. All this data shuffling is completed within the allotted time of two msec. BUFFR then checks the status of the various 7281 subchannels – turning them on or off as required, and again "hangs up" in a waiting loop for T00-msec time (T is the number of the next integral tenth second). Upon release from this waiting loop, the next program cycle is begun and the process continues as described in Sec. 8.1.1.

8.2 PROGRAM OPERATION FROM STANDPOINT OF DATA FLOW

The "guided tour" of a typical program cycle through which the reader was led in Sec. 8.1 was somewhat arduous, and there was, perhaps, a tendency to "lose sight of the forest through the trees." In this section, the operation of the program will be very briefly re-examined from quite a different aspect – that of the data flow through the program. It is hoped that the following, largely pictorial, description of program operation – from the point of view of the data flow through it – will aid in putting the operational description of the previous section in better perspective.

The data flow through the program is illustrated in Figs. 4 and 5. Figure 4 is a diagram of the main data flow through the program. The data flow illustrated starts with the raw input data from the sensors entering the program through the 7281 input subchannels. The flow then progresses as these input data are moved by BUFFR for recording on the output tape, and as the TRADEX and SKR input data are processed by the BUFFR, XTRAP, BETRAJ, and ACSM programs to be transformed into the TRADEX Target and Aircraft Track Files. The Acquisition Track Files are internally generated by the XTRAP and BETRAJ programs. These track file data are then fed to the sensor directing data computation subprograms where they are transformed into command data for the various sensors. The track file data are also fed to the plotting subprograms for display and to the re-entry prediction computation subprograms. The computed sensor command data are then fed back to the BUFFR program for recording on the output tape and for transmission through the 7281 output subchannels to the external sensor data links. The labeling of the data paths in Fig. 4 clearly indicates the data processing and handling functions performed by each of the major subprograms illustrated in that diagram.

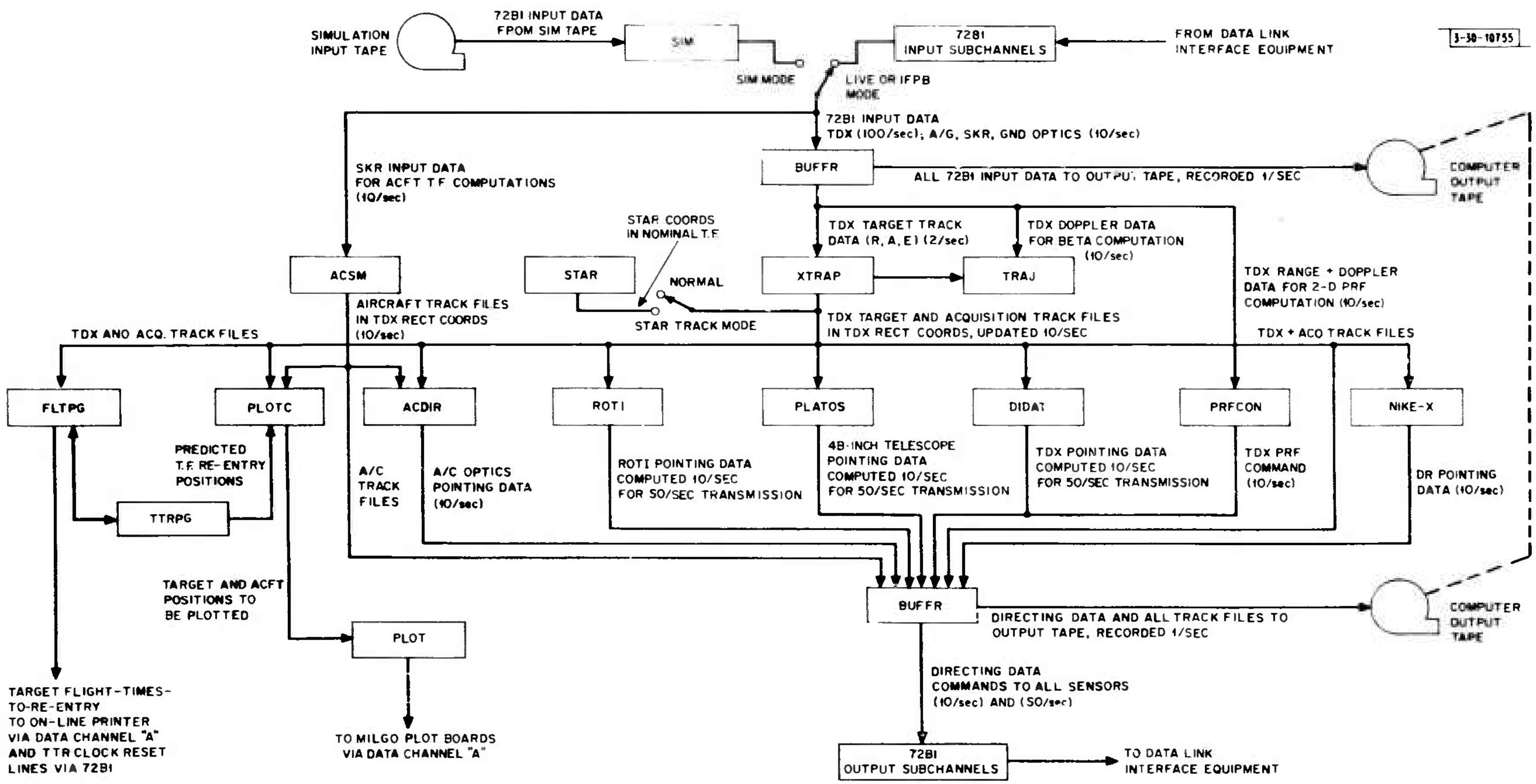

Fig. 4. Main internal data flow of RTP.

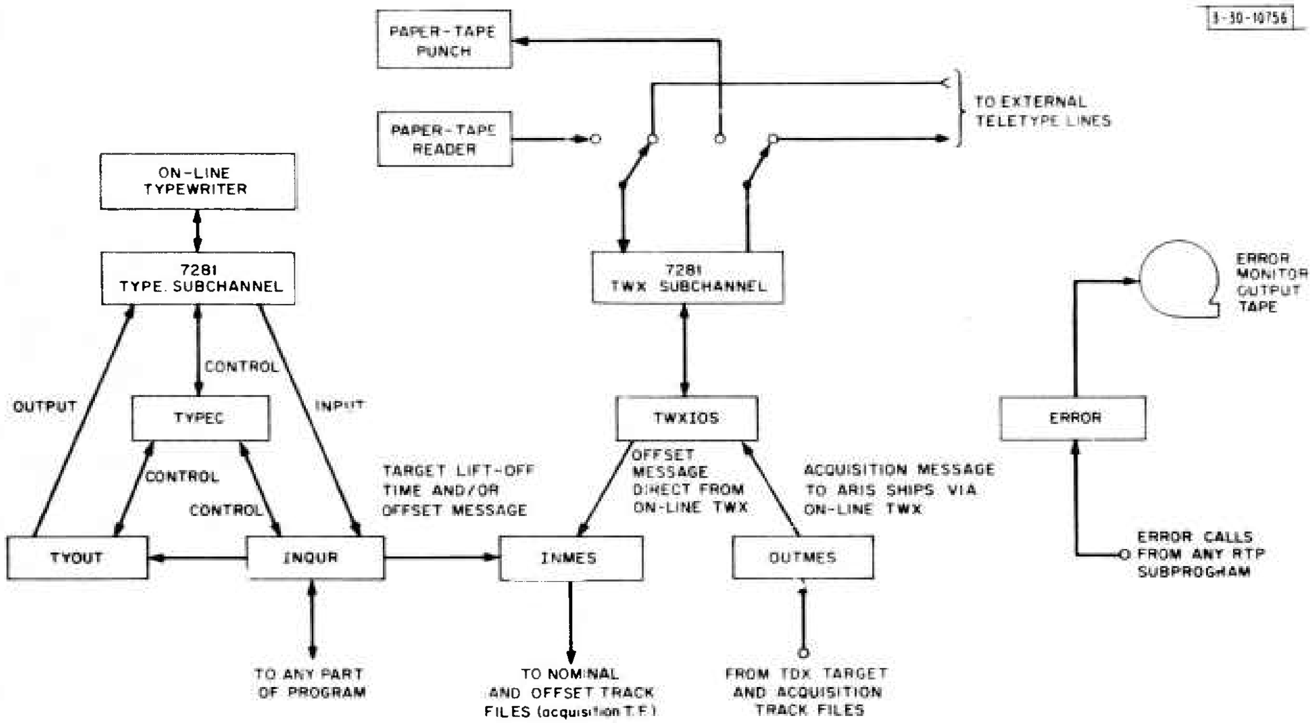

Fig. 5. RTP auxiliary internal data flow.

Figure 5 illustrates the auxiliary data flow through the program – the data communication flow between the program and the on-line typewriter and teletype connection, and the flow of data for recording on the Error Monitor Output Tape. The labeling of the data paths in Fig. 5 clearly indicates the data processing and handling functions performed by each of the subprograms in that diagram. The diagram itself is sufficiently simple and does not require any additional verbal explanation.

The separation of Figs. 4 and 5 indicates the difference between the main and auxiliary data flows. The main data flow is active every program cycle, while the auxiliary data flow is active only when either the on-line typewriter or teletype connection is in use, or when error data are to be recorded on the Error Monitor Output Tape. Note also that, although there is some linkage between the auxiliary and main data flows – principally through the acquisition track file data messages processed by INMES – in general, the auxiliary and main data flows are independent of one another. In the same way, the programs that service the typewriter, teletype, and error tape communications (as illustrated in Fig. 5), operate independently of those associated with the main data flow in Fig. 4.

APPENDIX

SUMMARY OF PRINCIPAL COMPUTATIONS PERFORMED BY THE PRESS RTP

1. SYMBOL DEFINITIONS

a major axis of Keplerian orbital ellipse in re-entry prediction computation, in ft.

e eccentricity of Keplerian orbital ellipse in re-entry prediction computation.

$f(\psi)$ magnitude of residual sea-level gravity acceleration component parallel to earth's axis at target geocentric latitude ψ.

$g(\psi)$ magnitude of principal sea-level gravity acceleration component directed toward earth center at target geocentric latitude ψ.

G reference acceleration of gravity; 32.1462 ft/sec^2.

h target altitude above mean sea level, ft; $h = r_c - R_\ell$.

L geodetic latitude of TRADEX location.

m number of points in track file.

M rotation matrix in re-entry prediction computation.

n mean anomaly of Keplerian orbital ellipse in re-entry prediction computation.

$\hat{q}$ unit vector in Keplerian orbital plane orthogonal to target position unit vector $\hat{r}_c$ in re-entry prediction computation.

r range to target from TRADEX origin. $r = \sqrt{x^2 + y^2 + z^2}$, ft.

$\dot{r}$ target range rate relative to TRADEX.

$\ddot{r}$ target range acceleration relative to TRADEX.

r_c distance from earth center to target, ft. $r_c = \sqrt{x_c^2 + y_c^2 + z_c^2}$.

$\hat{r}_c$ unit target position vector from earth center in re-entry prediction computation.

r_s target range from PRESS sensor.

R_c earth's equatorial radius.

R_ℓ local earth radius at position of target.

R_T local earth radius at TRADEX.

t time in seconds.

PRECEDING PAGE BLANK-NOT FILMED

u eccentric anomaly of target in Keplerian orbital ellipse in re-entry prediction computation.

V target velocity relative to earth, ft/sec; $v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$.

V_I inertial target velocity in re-entry prediction computation.

x, y, z target position components in TRADEX rectangular coordinates relative to TRADEX origin, ft: x, east; y, north; z, TRADEX vertical.

x_c, y_c, z_c target position components in TRADEX rectangular coordinates relative to earth center, ft: $x_c = x$, $y_c = y + \delta y_c$; $z_c = z + \delta z_c$.

$\dot{x}, \dot{y}, \dot{z}$ target velocity components in TRADEX rectangular coordinates, ft/sec.

$\ddot{x}, \ddot{y}, \ddot{z}$ total target acceleration components in TRADEX rectangular coordinates, ft/sec^2.

$\ddot{x}_g, \ddot{y}_g, \ddot{z}_g$ target ballistic acceleration components (drag deceleration excluded) in TARGET rectangular coordinates, ft/sec^2.

X_s, Y_s, Z_s coordinates of PRESS sensor location relative to TRADEX origin in TRADEX rectangular coordinates.

$\dot{X}_s, \dot{Y}_s, \dot{Z}_s$ velocity of PRESS sensor location in TRADEX rectangular coordinates.

α coefficient of first harmonic of earth gravitational potential; $\alpha = 0.00162$.

α_m position weighting coefficient in least-squares recursive smoothing algorithm.

β target ballistic coefficient $\frac{W}{C_D A}$, lb/ft^2.

γ_m velocity weighting coefficient in least-squares recursive smoothing algorithm.

δy_c y position of TRADEX origin relative to earth center; $\delta y_c = -22570$ ft

δz_c z position of TRADEX origin relative to earth center, $\delta z_c = 20923834$ ft.

ϵ eccentricity of earth ellipsoid; $\epsilon^2 = 0.006694$.

$\rho(h)$ atmospheric density, lbs/ft^3, obtained as a function of h from atmospheric property tables.

λ longitude of TRADEX location, positive east.

ψ geocentric latitude of target position.

θ true anomaly of target position in Keplerian orbital ellipse in re-entry prediction computation.

τ predicted flight time to re-entry, seconds, in re-entry prediction computation.

μ Newton's gravitational constant.

ω earth's rotational rate, 7.292×10^{-5} rad/sec.

SUPERSCRIPTS AND SUBSCRIPTS

quantity is measured or derived from a measurement.

c referenced to earth center.

I inertial.

m number of track file point.

^ least-squares estimate, also unit vector in re-entry prediction computation.

s relative to PRESS sensor.

2. BALLISTIC TRAJECTORY INTEGRATION

The integration equations are:

$$x(t + \Delta t) = x(t) + \dot{x}(t)\,\Delta t + \ddot{x}(t)\,\frac{\Delta t^2}{2} \quad , \tag{1}$$

similarly for y, z.

$$\dot{x}(t + \Delta t) = \dot{x}(t) + \ddot{x}(t)\,\Delta t \quad , \tag{2}$$

similarly for $\dot{y}, \dot{z}$.

The equations for the acceleration components used in (1) and (2) are given below. The argument t is omitted for clarity.

$$\ddot{x}_g = \omega^2 x_c + 2\omega(\dot{y}\sin L - \dot{z}\cos L) - \frac{x_c}{r_c}\,g(\psi)\left(\frac{R_e}{r_c}\right)^2 \tag{3}$$

$$\ddot{y}_g = \omega^2 \sin L\,(y_c \sin L - z_c \cos L) - 2\omega\dot{x}\sin L - \frac{y_c}{r_c}\,g(\psi)\left(\frac{R_e}{r_c}\right)^2 - f(\psi)\left(\frac{R_e}{r_c}\right)^2 \cos L \tag{4}$$

$$\ddot{z}_y = \omega^2 \cos L\,(z_c \cos L - y_c \sin L) + 2\omega\dot{x}\cos L - \frac{z_c}{r_c}\,g(\psi)\left(\frac{R_e}{r_c}\right)^2 - f(\psi)\left(\frac{R_e}{r_c}\right)^2 \sin L \quad , \tag{5}$$

where

$$x_c = x$$

$$y_c = y + \delta y_c$$

$$z_c = z + \delta z_c$$

and

$$\left.\begin{aligned} \ddot{x} &= \ddot{x}_g - \frac{\rho V}{2\beta}\,\dot{x} \\ \ddot{y} &= \ddot{y}_g - \frac{\rho V}{2\beta}\,\dot{y} \\ \ddot{z} &= \ddot{z}_y - \frac{\rho V}{2\beta}\,\dot{z} \end{aligned}\right\} \tag{6}$$

The target geocentric latitude ψ is computed from

$$\psi = \sin^{-1}\left(\frac{y_c \cos L + z_c \sin L}{r_c}\right) . \tag{7}$$

The gravity potential components $g(\psi)$, $f(\psi)$ are computed from

$$g(\psi) = G\left[1 + \alpha\left(\frac{R_e}{r_c}\right)^2 (1 - 5 \sin^2 \psi)\right] \tag{8}$$

$$f(\psi) = 2G\alpha\left(\frac{R_e}{r_c}\right)^2 \sin \psi . \tag{9}$$

3. TRACK FILE ALTITUDE COMPUTATIONS

$$h = r_c - R_\ell \tag{10}$$

$$r_c = \sqrt{x_c^2 + y_c^2 + z_c^2} \tag{11}$$

$$R_\ell = R_e \sqrt{\frac{1 - \epsilon^2}{1 - \epsilon^2 \cos^2 \psi}} . \tag{12}$$

When the z coordinate of an Aircraft Track File is computed from the aircraft altimeter measurement in lieu of SKR elevation, the relation for z^* (before smoothing) is:

$$z^* = h^* + R_\ell - R_T\left(\frac{r^{*2} - (h^* + R_\ell - R_T)^2}{2R_T}\right) , \tag{13}$$

where h^* is the altimeter altitude measurement averaged over one second, and r^* is the SKR aircraft range measurement averaged over one second.

4 DRAG DECELERATION COMPUTATION

The drag deceleration computation is performed when the altitude of the target being tracked is below 400 kft (in atmospheric re-entry). The purpose of this computation is to estimate the magnitude of the drag deceleration on the target for use in computing the total target acceleration components [see Eq. (6)]. The basic equation used to compute the drag estimate from the TRADEX doppler tracking measurements is given below.

$D = \frac{\rho V}{2\beta}$ is the drag deceleration coefficient.

$$D = \frac{\rho V}{2\beta} = \frac{\dfrac{x\ddot{x}_g + y\ddot{y}_g + z\ddot{z}_g + v^2 - \dot{r}^2}{r} - \hat{\ddot{r}}^*}{\hat{\dot{r}}^*} . \tag{14}$$

$\dot{r}^*$, $\ddot{r}^*$, are obtained from a recursive least-squares parabolic fit of the TRADEX target doppler measurements over a time span ranging from 0.8 to 2.5 sec. The other variables in Eq. (14) are obtained from the Target Track File. The value $(\rho V/2\beta)$ computed from (14) is not used in (6) directly. The values computed from (14) are first smoothed further and corrected for doppler tracking biases before being applied in (6); however, the details of the additional massaging applied to $\rho V/2\beta$ are beyond the scope of the present discussion.

The recursive least-squares parabolic fitting algorithm used to smooth the TRADEX doppler measurements for estimation of $\hat{\ddot{r}}^*$ and $\hat{\dddot{r}}^*$ is given below.

$$\hat{\dot{r}}_{n+1} = \hat{\dot{r}}_{n+1/n} + \alpha_{n+1}(\dot{r}^*_{n+1} - \hat{\dot{r}}_{n+1/n}) \tag{15}$$

$$\hat{\ddot{r}}_{n+1} = \hat{\ddot{r}}_{n+1/n} + \gamma_{n+1}(\dot{r}^*_{n+1} - \hat{\dot{r}}_{n+1/n}) \tag{16}$$

$$\hat{\dddot{r}}_{n+1} = \hat{\dddot{r}}_{n+1/n} + \nu_{n+1}(\dot{r}^*_{n+1} - \hat{\dot{r}}_{n+1/n}) \quad . \tag{17}$$

$\dot{r}^*_n$ is the nth TRADEX doppler measurement sample.

$$\hat{\dot{r}}_{n/n+1} = \hat{\dot{r}}_n + \hat{\ddot{r}}_n k\Delta t + \hat{\dddot{r}}_n(k\Delta t)^2/2 \tag{18}$$

$$\hat{\ddot{r}}_{n/n+1} = \hat{\ddot{r}}_n + \hat{\dddot{r}}k\Delta t \tag{19}$$

$$\hat{\dddot{r}}_{n/n+1} = \hat{\dddot{r}}_n \tag{20}$$

$$\alpha_n = \frac{3(3n^2 - 3n + 2)}{n(n+1)(n+2)} \quad n \geqslant 1 \tag{21}$$

$$\gamma_n = \begin{cases} 0 & ; \quad n = 1 \\ \dfrac{1}{\Delta t} & ; \quad n = 2 \\ \dfrac{18(2n-1)}{n(n+1)(n+2)\Delta t} & ; \quad n \geqslant 3 \end{cases} \tag{22}$$

$$\nu_n = \begin{cases} 0 & ; \quad n = 1 \\ \dfrac{60}{n(n+1)(n+2)\Delta t^2} & ; \quad n \geqslant 3 \end{cases} \tag{23}$$

$$\Delta t = t_{m+1} - t_m \quad .$$

The value of n above is truncated at N_{max} – a value corresponding to a data span between 0.8 and 2.5 sec. The value of N_{max} is chosen to minimize the error in the estimate due to the combination of noise in the data and biases in the estimate resulting from higher order behavior of the data than second degree.

5. TRACK FILE SMOOTHING COMPUTATION

The following recursive smoothing algorithm generates a least-squares trajectory bit to the TRADEX target tracking measurements, the resulting position and velocity coordinate estimates being placed into the TRADEX Target Track Files. Essentially, the same algorithm is used to smooth the SKR tracking measurements and aircraft altimeter measurements to form the Aircraft Track Files.

First, the measured target range, azimuth, and elevation data are converted from radar coordinates to TRADEX rectangular coordinates as follows:

$$\left.\begin{aligned} x^*_m &= r^*_m \cos El^*_m \sin Az^*_m \\ y^*_m &= r^*_m \cos El^*_m \cos Az^*_m \\ z^*_m &= r^*_m \sin El^*_m \end{aligned}\right\} \quad . \tag{24}$$

The converted measurements are then individually smoothed in rectangular coordinates to obtain position and velocity estimates.

$$\hat{x}_m(t_m) = \hat{x}_{m-1}(t_m) + \alpha_m[x^*_m - \hat{x}_{m-1}(t_m)] \quad , \tag{25}$$

similarly for $\hat{y}_m$, $\hat{z}_m$.

$$\hat{\dot{x}}_m(t_m) = \hat{\dot{x}}_{m-1}(t_m) + \gamma_m[x^*_m - x_{m-1}(t_m)] \quad , \tag{26}$$

similarly for $\hat{\dot{y}}_m$, $\hat{\dot{z}}_m$.

$\hat{x}_{m-1}(t_m)$ = track file position component estimate obtained at t_{m-1} and extrapolated ahead to time t_m. (Track file position component after trajectory integration step, but before new data point x^*_m is included).

$\hat{x}_m(t_m)$ = track file position component estimate at time t_m.

$\hat{\dot{x}}_{m-1}(t_m)$ = track file velocity component estimate obtained at time t_{m-1} and extrapolated ahead to time t_m. (Track file velocity component after trajectory integration step but before new data point x^*_m is included).

$\hat{\dot{x}}_m(t_m)$ = track file velocity component estimate at time t_m.

$$\alpha_m = \frac{2(2m-1)}{m(m+1)} \quad . \tag{27}$$

$$\gamma_m = \frac{6}{m(m+1)\,\Delta t} \quad ; \quad m \geq 2 \quad . \tag{28}$$

$$\Delta t = t_m - t_{m-1} \quad . \tag{29}$$

The value of m in the above equations is presently truncated when m reaches 100 in the case of the TRADEX Target Track Files. Since TRADEX data are sampled twice a second, this corresponds to a maximum smoothing span of 50 sec.

In the case of the Aircraft Track Files, the data smoothed by this algorithm are sampled once a second. The data sample applied is not raw data but a simple arithmetic average of the raw data over the previous second. In this case, the value of m is truncated when m reaches 10, which corresponds to a maximum smoothing span of 10 sec.

6. RE-ENTRY PREDICTION COMPUTATION

The predicted position and time of re-entry are computed by assuming that the ballistic trajectory of a target may be approximated by a Keplerian orbital ellipse. The parameters of the ellipse are computed from the current position and velocity estimates in the Target Track

File. The predicted position of re-entry is that point on the ellipse which is 300 kft above the earth's surface (the intersection of the ellipse with a sphere coencentric with the earth and 300 kft above it at TRADEX).

The basic parameters that define the shape of the ellipse are its major axis, a, and its eccentricity, e.

The major axis, a, may be computed from the inertial velocity and distance of the target from the earth's center by the following relation:

$$a = \frac{\mu}{2\left(\frac{\mu}{r_c} - \frac{V_I^2}{2}\right)} \tag{30}$$

where

$$r_c = \sqrt{x_c^2 + y_c^2 + z_c^2}$$

$$V_I = \sqrt{\dot{x}_I^2 + \dot{y}_I^2 + \dot{z}_I^2}$$

where

$$\dot{x}_I = \dot{x} - y_c\omega \sin L + z_c\omega \cos L$$

$$\dot{y}_I = \dot{y} + x_c\omega \sin L$$

$$\dot{z}_I = \dot{z} - x_c\omega \cos L \quad .$$

The eccentricity, e, may be found from the following relation:

$$e^2 = \left(\frac{r_c}{a} - 1\right)^2 + \frac{(r_c\dot{r}_c)^2}{\mu a} \quad , \tag{31}$$

where the product $r_c\dot{r}_c$ may be computed directly from:

$$r_c\dot{r}_c = x_c\dot{x}_I + y_c\dot{y}_I + z_c\dot{z}_I \quad . \tag{32}$$

A convenient way to characterize the position of the target along the orbital ellipse for purposes of the time-to-re-entry computation is in terms of the eccentric anomaly, u. The eccentric anomaly, u, is related to r_c along the ellipse by the relation:

$$r_c = a(1 - e \cos u) \tag{33}$$

where the major axis, a, has been found from (30) and the eccentricity, e, is determined by (31).

The eccentric anomaly of the current target position, u_1, is found from

$$u_1 = \tan^{-1}\left\{\frac{r_c\dot{r}_c}{\sqrt{\mu a}}\left(\frac{a}{a - r_c}\right)\right\} \quad . \tag{34}$$

The eccentric anomaly of the predicted re-entry position, u_2, is found directly from (33).

$$u_2 = \cos^{-1}\left\{\frac{1}{e}\left[1 - \frac{(R_T + 300{,}000)}{a}\right]\right\} \quad . \tag{35}$$

The predicted target flight time, τ, from the current position to re-entry is then given by:

$$\tau = \frac{u_2 - u_1 - e(\sin u_2 - \sin u_1)}{n} \quad , \tag{36}$$

where n is the mean anomaly,

$$n = \frac{\sqrt{\mu a}}{a^2} \quad .$$

To obtain the predicted re-entry position, a more convenient way of characterizing the orbital ellipse is by means of the "true anomaly," Θ. The true anomaly, Θ, is the angle measured at the earth center in the direction of target motion between the point of perihelion of the ellipse and the position of the target. The characterization of target motion on the ellipse in terms of the true anomaly is given by:

$$r_c = \frac{a(1 - e^2)}{1 + e\cos\Theta} \quad . \tag{37}$$

Thus, the true anomaly, Θ, of the current position is given by:

$$\Theta_1 = \cos^{-1}\left\{\frac{1}{e}\left[\frac{a(1 - c^2)}{r_c} - 1\right]\right\} \quad , \tag{38}$$

and the true anomaly, Θ_2, at the predicted re-entry position is given by:

$$\Theta_2 = \cos^{-1}\left\{\frac{1}{e}\left[\frac{a(1 - c^2)}{R_T + 300{,}000} - 1\right]\right\} \quad . \tag{39}$$

Now, define

$$\varphi = \Theta_2 - \Theta_1 \quad . \tag{40}$$

φ is the angle measured at the earth center in the elliptical orbit plane between the current target position and the predicted re-entry position.

The unit vector from the earth center to the current target position is $\hat{r}_c$. Next, consider the unit vector $\hat{q}$ in the orbit plane orthogonal to $\hat{r}_c$ and pointed in the same sense as the motion of the target in the ellipse. $\hat{q}$ is given by the following relation:

$$\hat{q} = \frac{\vec{V}_1 - (\vec{V}_1 \cdot \hat{r}_c)\hat{r}_c}{|\vec{V}_1 - (\vec{V}_1 \cdot \hat{r}_c)\hat{r}_c|} \quad , \tag{41}$$

where $\vec{V}_I$ is the inertial velocity vector composed of components $(\dot{x}_1, \dot{y}_1, \dot{z}_1)$, and $(\vec{V} \cdot \hat{r}_c)$ is the scalar product of $\vec{V}$ and $\hat{r}_c$.

The predicted re-entry position vector relative to the earth center is then given by:

$$\vec{r}_{pI} = (R_T + 300{,}000)(\hat{r}_c \cos\varphi + \hat{q}\sin\varphi) \quad . \tag{42}$$

Now, $\vec{r}_{pI}$ is the predicted re-entry position vector on the ellipse fixed in inertial space. However,

we wish to obtain the predicted position relative to the earth. Consequently, since the earth rotates under the orbital ellipse as the target travels along its trajectory, it is necessary to rotate the vector $\vec{r}_{pI}$ by the amount of the earth's rotation during the predicted time-of-flight to re-entry τ computed from Eq. (36). Thus, the predicted re-entry position vector $\vec{r}_p$, measured from the earth center and relative to a fixed earth, is given by:

$$\vec{r}_p = M\vec{r}_{pI} \quad , \tag{43}$$

where $\vec{r}_{pI}$ is the result of (42) and M is the requisite rotation matrix given below.

$$M = \begin{Bmatrix} \cos\omega\tau & \sin L \sin\omega\tau & -\cos L \sin\omega\tau \\ -\sin L \sin\omega\tau & 1 - \sin^2 L(1-\cos\omega\tau) & \sin L \cos L(1-\cos\omega\tau) \\ \cos L \sin\omega\tau & \sin L \cos L(1-\cos\omega\tau) & 1-\cos^2 L(1-\cos\omega\tau) \end{Bmatrix} \quad . \tag{44}$$

7. SENSOR DIRECTING DATA COMPUTATIONS

The pointing commands computed for the various sensors usually consist of range, azimuth, and elevation of the target from the sensors, and the rates of these coordinates. A slit-roll command is also computed for the airborne optics sensors.

The first step in the directing data computation for each sensor is the translation of the target position and velocity from the TRADEX origin to the position and velocity of the sensor location. However, only the airborne optics sensors have a nonzero sensor location velocity with respect to the TRADEX origin. Consequently, the translation of target velocity is only performed in the case of the airborne sensors. The velocity of the airborne sensor location is given by the velocity components in the associated Aircraft Track File. The general position and velocity translation equations are:

$$X_s = x - X_s \tag{45}$$

$$Y_s = y - Y_s$$

$$Z_s = z - Z_s$$

$$\dot{X}_s = \dot{x} - \dot{X}_s$$

$$\dot{Y}_s = \dot{y} - \dot{Y}_s \tag{46}$$

$$\dot{Z}_s = \dot{z} - \dot{Z}_s \quad ,$$

where X_s, Y_s, Z_s; $\dot{X}_s, \dot{Y}_s, \dot{Z}_s$ are the position and velocity of the sensor location with respect to the TRADEX origin.

Following the translation process, the target coordinates are extrapolated to compensate for the data transmission delay to the sensor:

$$X_s = X_s + \dot{X}_s \Delta t + \ddot{x}\Delta t^2/2$$

$$\dot{X}_s = \dot{X}_s + \ddot{x}\Delta t \quad . \tag{47}$$

Similarly, for $Y_s, \dot{Y}_s$; $Z_s, \dot{Z}_s$.

Next, if the sensor is located more than one-half mile from TRADEX, the target coordinates are rotated from the TRADEX coordinate system into the corresponding coordinate system whose Z axis is the local vertical at the sensor location. This rotation compensates for the coordinate system change due to the earth curvature. The equations are:

$$\begin{Bmatrix} X_s \\ Y_s \\ Z_s \end{Bmatrix} = M \begin{Bmatrix} X_s \\ Y_s \\ Z_s \end{Bmatrix} \tag{48}$$

$$\begin{Bmatrix} \dot{X}_s \\ \dot{Y}_s \\ \dot{Z}_s \end{Bmatrix} = M \begin{Bmatrix} \dot{X}_s \\ \dot{Y}_s \\ \dot{Z}_s \end{Bmatrix} \quad , \tag{49}$$

where M is the requisite 3 × 3 rotation matrix that converts the target coordinates from the orientation of the TRADEX rectangular coordinate system to that of the local rectangular coordinate system at the sensor location. In the case of the aircraft sensors, the local rectangular coordinate system is aligned with the orientation of a stable platform carried by the aircraft.

Finally, following the coordinate rotation where applicable, the sensor pointing commands and command rates are computed from the extrapolated and converted target position and velocity components in the sensor rectangular coordinate system according to the equations given below. Of course, not all the commands and rates described below are computed for every sensor; for example, only azimuth, elevation, and roll commands are computed for the airborne optics sensors.

Range:

$$r_s = \sqrt{x_s^2 + y_s^2 + z_s^2} + \Delta_r(El_s, r_s)^* \quad . \tag{50}$$

Range Rate:

$$\dot{r}_s = \frac{x_s\dot{x}_s + y_s\dot{y}_s + z_s\dot{z}_s}{r_s} \quad . \tag{51}$$

Range Acceleration:

$$\ddot{r}_s = \frac{\dot{x}_s^2 + \dot{y}_s^2 + \dot{z}_s^2 - \dot{r}_s^2 + x_s\ddot{x}_s + y_s\ddot{y}_s + z_s\ddot{z}_s}{r_s} \quad . \tag{52}$$

Azimuth:

$$Az_s = \tan^{-1}\frac{x_s}{y_s} \quad . \tag{53}$$

Elevation:

$$El_s = \tan^{-1}\frac{z_s}{\sqrt{x_s^2 + y_s^2}} + \Delta_E(El_s, r_s)^* \quad . \tag{54}$$

---

*$\Delta_r(El_s, r_s)$, $\Delta_E(El_s, r_s)$ are the range and elevation corrections, respectively, for atmospheric refraction.

Azimuth Rate:

$$\dot{Az}_s = \frac{\dot{x}_s y_s - x_s \dot{y}_s}{x_s^2 + y_s^2} \quad . \tag{55}$$

Elevation Rate:

$$\dot{El}_s = \frac{\dot{z}_s \sqrt{x_s^2 + y_s^2} - (x_s \dot{x}_s + y_s \dot{y}_s) \dfrac{z_s}{\sqrt{x_s^2 + y_s^2}}}{x_s^2 + y_s^2 + z_s^2} \quad . \tag{56}$$

Slit Roll Angle:

$$\text{Roll} = \tan^{-1} \left(\frac{-\dot{El}_s}{\dot{Az}_s \cos El_s} \right)$$

$$= \tan^{-1} \left[- \frac{\dot{z}_s (x_s^2 + y_s^2) - z_s (x_s \dot{x}_s + y_s \dot{y}_s)}{\sqrt{x_s^2 + y_s^2 + z_s^2}\ (\dot{x}_s y_s - x_s \dot{y}_s)} \right] \quad . \tag{57}$$

8. STAR TRACK COMPUTATIONS

The Sidereal Hour Angle (SHA) and Declination (Dec) of the star to which the optics sensors are to be pointed, and the Greenwich Hour Angle of Aries (GHAγ) at the beginning and end of the current day, are given as input data to the program. At any instant, the current value of GHAγ is obtained by interpolating between the two given values to the current time of day.

The Local Hour Angle (LHA) of the star at any instant is computed from the GHAγ at that instant, the SHA of the star, and the longitude of TRADEX by:

$$LHA = SHA - GHA\gamma + \lambda \quad . \tag{58}$$

The azimuth and elevation of the star at TRADEX are then given by

$$Az = \tan^{-1} \left[\frac{\cos(Dec) \sin(LHA)}{\sin(Dec) \cos L - \cos(Dec) \sin L \cos(LHA)} \right] \tag{59}$$

$$El = \sin^{-1} \left[\cos(Dec) \cos L \cos(LHA) + \sin(Dec) \sin L \right] \quad , \tag{60}$$

where L, λ are the geodetic latitude and longitude, respectively, of the TRADEX origin.

The position components of the star in TRADEX rectangular coordinates are then computed from:

$$\left. \begin{aligned} x &= 10^9 \sin Az \cos El \\ y &= 10^9 \cos Az \cos El \\ z &= 10^9 \sin El \end{aligned} \right\} \quad , \tag{61}$$

where 10^9 ft is the dummy range of the star – a practical approximation for infinity – and Az and El are obtained from (61) and (62).

Finally, the velocity components of the star in TRADEX rectangular coordinates, due to earth rotation, are computed from:

$$\left.\begin{aligned} \dot{x} &= x \cos L \\ \dot{y} &= -x \sin L \\ \dot{z} &= (y \sin L - x \cos L) \end{aligned}\right\} . \qquad (62)$$

When in Star Track Mode, the values of $x, y, z, \dot{x}, \dot{y}, \dot{z}$ computed above are placed into the Nominal Track File, which, when selected as the Directing Track File, will cause the optics sensors to be pointed at the designated star.